your computer

AUGUST, 1983 $2.00* NZ$3

FOR BUSINESS AND PLEASURE

BONUS LIFTOUT your BUSINESS computer

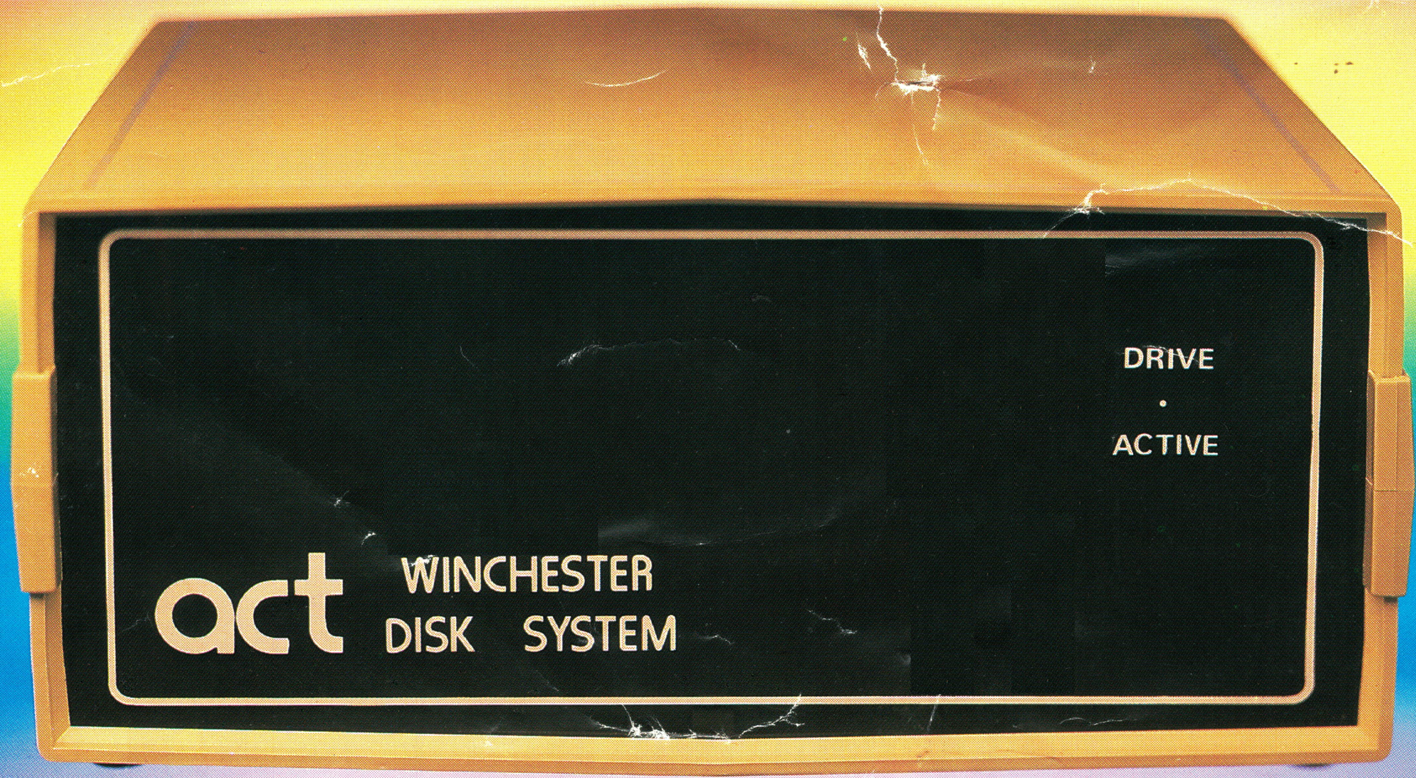
CASH & CARRY COMPUTERS

The Under-$500 Machines • VZ200 • SpectraVideo • VIC 20 • MicroBee • COM-X • TI 99/4A • MicroProfessor II • Wizzard • Spectrum • dBase, Assembler Tutorials • Networks Analysed

ISSN 0725-3931

Registered for posting as a publication — Publication number NBP4384

We Have The Answer To Your Storage Problems



Take a hard line on software with the ACT WINCHESTER HARD DISK SYSTEM

THE POWERFUL ACT

The ACT Hard Disk System will upgrade your computer to a powerful, high performance machine. Supported throughout Australia, the ACT Hard Disk will give you fast convenient data storage.

THE RELIABLE ACT

Far greater storage than floppy drives with reliable performance under all conditions, makes the ACT Hard Disk unbeatable value. Designed and manufactured in Australia.

5 — 25 Megabytes of Hard Disk Storage.
5 Plus 5 Fixed/Removable Hard Disk.
5 Megabytes Removable Hard Disk Cartridge.

THE LOGICAL ACT

APPLE II: DOS 3.3, PASCAL & CP/M
IBM PC: PC DOS & CP/M86.
OSBORNE — KAYPRO — SIGMA OKI — TRS80 II & III — SANYO — NORTHSTAR.

HEATH/ZENITH — XEROX 820 — S100
MR FLOPPY PTY LTD.

SOON FOR THE DEC RAINBOW & SIRIUS.

THE SHAREABLE ACT — PC NET

PC Net is a hard disk sharing system — simple and cost effective. The disk is divided into logical drives, allowing independent users to share the disk. Available for most popular brands of microcomputer.

W.A.
MICRO E.D.P. HARDWARE,
9a/1 Leura Avenue,
Claremont, WA 6010
Phone: (09) 384 5511

N.S.W.
act
75 Willoughby Road,
Crows Nest, NSW 2065
Phone: (02) 439 6300
Telex: AA 24816

S.A.
COMPUTER MART PTY LTD,
123 Gouger Street,
Adelaide, SA 5000
Phone: (08) 51 5363, 212 2888

DEALER ENQUIRIES WELCOME

inside your computer

Vol 3, No 1.
August, 1983.

Audited by Audit Bureau of Circulations abc
for period Oct. 1 '82 to Mar. 31 '83
AVERAGE NET PAID SALES
per regular issue
IN EXCESS OF
15,500
excluding unsold and free copies

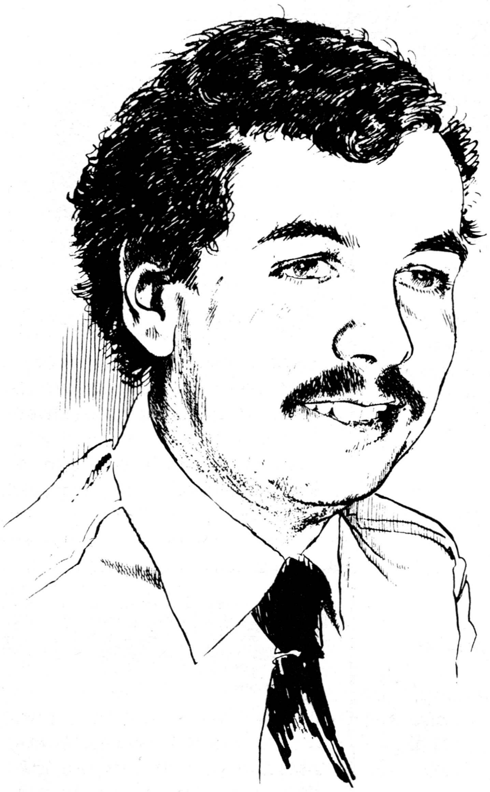
Here's the answer for everyone who has asked us to look at the booming small home computer market – a review of just about everything available, starting on Page 20.

REGULAR PULL-OUT SECTION
AUGUST 1983
YOUR BUSINESS COMPUTER MAGAZINE

And if it's the business end of the market you're interested in you've come to the right place: this month we launch Your Business Computer. **See the centre pages.**

dBase II is one of the business world's most popular programs, but it can be hard on the new user. We set out to solve that with a tutorial series – see page 74 for this month's instalment.

editorial



WASN'T IT Humpty Dumpty who said, "Whenever I say something, it means what I want it to mean – no more and no less"?

Because of changes in the microcomputer industry and its markets, I think it's time I made it clear just what I mean by "personal computer" and "microcomputer". The two are different – at least, they are in my vocabulary.

A personal computer (PC) is a microprocessor-based machine with integrated video circuitry (or even screen) and keyboard, which is dedicated to use by a single person, and the software for which reflects the needs of a single person and not an organisation.

By contrast, a microcomputer has one or more external terminals. Its software may include personal applications – for example, SuperCalc – but often it is used for organisation-orientated tasks, such as general accounting.

For example, the IBM-PC is a personal computer, as are the Apple, Tandy Model III and others. However, the Godbout, Morrow, Universe, Plexus and the like are microcomputers. Some machines, like the Tandy Model 16 and the NEC APC, could fall in either camp.

This is a useful distinction to make. The more experienced of our readers will appreciate that there is more to this than meets the eye. Our present and potential contributors would, I'd like to think, appreciate these definitions and stick to this standard, as it will be helpful to our readers.

On the subject of definitions, would the sellers of 68000-based systems please make up their minds whether their computers are 16- or 32-bit machines? They are the first to denigrate the 8088 (inside the IBM-PC and others) as being "only eight-bit, really" (it's 16-bit internally, eight-bit externally) while describing their own systems as 32-bit (they're to some extent 32-bit internally, and certainly only 16-bit externally).

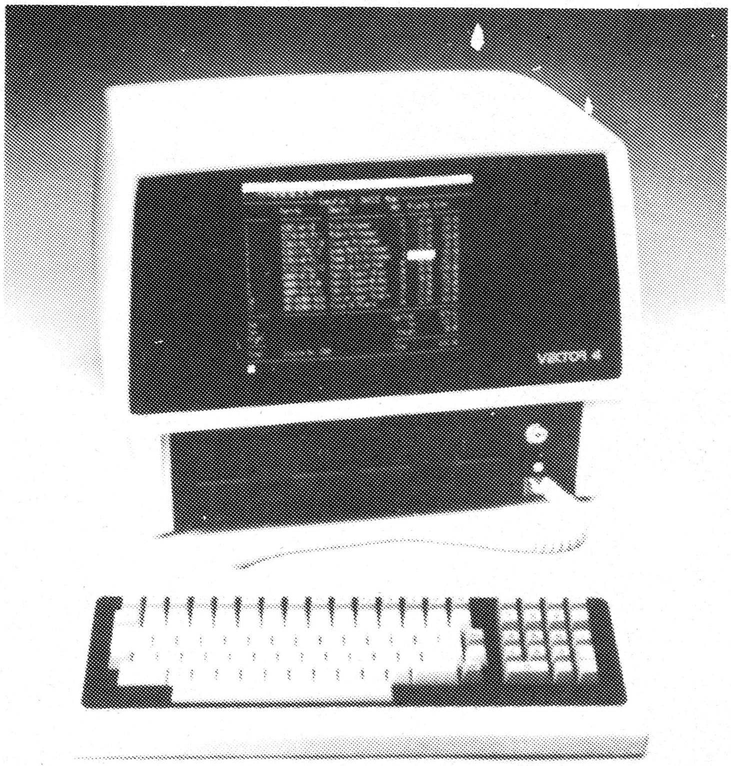
These people would tell you the dear old 8080 is a 16-bit machine by their lights. This set of double standards is a gigantic pain in the bum, and they're unlikely to get good press write-ups as long as they continue this hype.

Sure, the 68000 is a good processor, but word length – even if they were correct in their statements – is not the be-all and end-all of processor performance. Until these companies get their act together, they will continue to be their own (and their customers') worst enemies.

– Les Bell

YOUR COMPUTER is published monthly by the Federal Publishing Company Proprietary Limited. Printed by ESN-The Litho Centre, Waterloo NSW (phone (02) 662-8888). **Offices:** Editorial and advertising (except Victoria), 140 Joynton Avenue, Waterloo NSW 2017. Phone: (02) 663-9999. Postal Address: PO Box 21, Waterloo NSW 2017. Telex: FEDPUB AA74488. **Victorian Advertising:** Mel Godfrey, 150 Lonsdale St, Melbourne 3000. Phone: (03)662-1222. Telex: FEDPUB AA34543. **Publisher:** Michael Hannan. Typeset by Hughes Phototype, Cremorne NSW, (02) 90-4034. Distributed nationally by Gordon and Gotch. Registered for posting as a publication – Publication No. NBP4384. See back page for information on copyright, contributions, reader services and subscriptions. **Source Mail:** Les Bell TCY314; Matt Whelan TCY267. **MICC Bulletin Board:** (02) 662-1686 – Les Bell 1085, Matt Whelan 1, or leave a message in comments. *Recommended and maximum price only. ISSN 0725-3931.

your computer news



The Vector 8/16-bit microcomputer.

Lots Of News From Dicker Data

DICKER DATA Projects has introduced an 'entry level' version of the popular Vector 4 8/16-bit business computer. Called the Vector 4/10, it has a single integrated 13 cm floppy disk drive with 630K capacity as well as 128K of user memory.

Included as standard equipment on the Vector 4/10 are CP/M and CP/M-86 operating systems, GSX-86 graphics software, Scope (a full-screen editor), Raid (assembly language debugger), ZSM assembly language, BASIC-80, Memorite III word processing and Execuplan II financial modelling.

Priced at $4295, it can be used as a stand-alone computer or as a workstation on Vector's recently announced LINC local area network.

Dicker Data has also released MaiLINC, an electronic mail system for LINC. MaiLINC enables LINC users to compose a message and send it to another user's 'mailbox' - all on the monitor screen. The system allows the user to switch back and forth between sending and receiving messages, making it easy to respond to each 'letter' immediately.

Dicker Data has just completed negotiations with Compact Australia to supply Compact Accounting Software on all Vector computers. Compact incorporates a full suite of accounts software; all models can operate on a stand-alone basis, but when integrated cover debtors, creditors, general ledger, inventory and order processing/invoicing. The software can be operated under both CP/M and CP/M-86, making it very versatile.

The first Dicker Data Business Computer Centre opened at 261 George St, Sydney, recently – planned to be only the first of many. The centre will display the complete Vector range of hardware and software, along with computer books, magazines and consumables.

For further information on all these Dicker Data matters, contact the company on (02)525-2122. □

New Products From Commodore

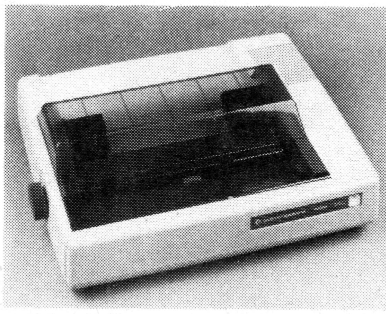
COMMODORE HAS introduced a new range of accessories to team with the popular VIC-20 and 64 computers.

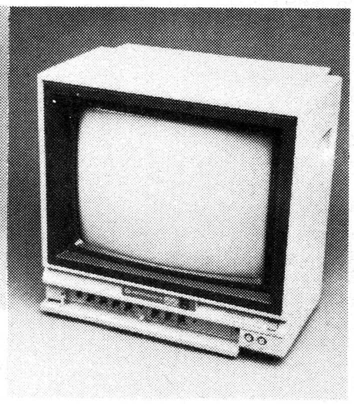
First release was the 1520 printer/plotter, a four-colour printer that retails for $360. This was followed by the VIC 1525 graphic printer, an economical 80-column dot matrix printer with graphics capability, retailing at $479, and the 1526 quality printer, retailing for $629. The 1526 not only prints double-size characters (as does the 1525), but also has an enhance command enabling the size of these letters to be further increased.

Commodore has also released the 1701 colour monitor, a high-resolution, high-quality monitor which will retail in Australia for around $400.

Commodore International, the parent company of Commodore Business Machines in Australia, recently reported another record-breaking quarter for its global operations, boosted largely by the success of the Commodore 64 and VIC-20 home computers. According to Roger Davis, National Sales Manager for Australia, the company has a target sales figure of 100,000 VIC-20s for 1983, and market response so far has indicated it will easily reach this figure.

For more information on Commodore products, contact Commodore Computer, 5 Orion St, Lane Cove 2066. (02)427-4888. □





Commodore's 1526 quality printer (left), and the new 1701 colour monitor.

Steps Of Hope For Paraplegics

JAN BURGESS, former international gymnast, is seen taking her first tentative steps after 14 years in a wheelchair, crippled by an accident on a trampoline which left her paralysed from the waist down. She is learning to walk again using a computerised device, developed by British engineer Dr Hugh Grenfell (left), to provide external electrical stimulation to her leg and hip muscles.

Powered by two six-volt batteries, the analogue computer is linked to hand-operated command switches and electrodes attached to three groups of muscles: the quadriceps, for standing, the iliopsoas, to raise the leg, and the dorsiflexors to raise the foot.

The patient can adjust the power and frequency of the stimulations with controls on the plastic boxes used to house the prototype. With further development Dr Grenfell expects to include all the wires and circuitry in a small pack to be carried on a belt or shoulder strap, eliminating the need for the walking frame.

Jan Burgess, 34, who represented Wales before the accident, became the guinea-pig for Dr Grenfell's invention in October 1982. She went on a special muscle-building diet and

began daily exercises to strengthen the muscles which had wasted away during her years confined to a wheelchair. She has now managed up to 24 steps at a time.

Dr Grenfell, a former chief research engineer with the Steel Company of Wales, has formed his own company to develop this and other aids for the handicapped. He hopes that thousands of people crippled in accidents will be able to benefit from the device and he intends to try it with people paralysed by illness, like muscular dystrophy, and stroke victims.

For further information contact the Information Officer, British Consulate-General, Gold Fields House, Sydney 2000. (02)27-7521. □

Dr Hugh Grenfell helping Jan Burgess operate his computerised walking aid for people paralysed by accident or illness.

Small Business Computer Conference

SMALL BUSINESS operators considering the purchase of a computer system will find help at the 1983 Small Business Computer Conference, to be held at the Frankston Campus of the Chisholm Institute of Technology.

The conference, scheduled to run over two and a half days from August 15-17, will feature lectures and demonstrations from speakers representing a cross-section of the computer industry, with a particular focus on what computers can do, rather than the technical aspects of how they work.

The theme for the conference is 'Computing – Your Business', emphasising the changes small businesses face in adjusting to computerisation.

Running concurrently with the lecture program will be a representative exhibition, featuring many of the personal computer systems suitable for small businesses, and there will also be a series of 'hands-on' workshops for conference delegates who want first-hand experience with a small computer system.

The conference is being presented by the Division of Continuing Education, Chisholm Institute of Technology (Frankston Campus), in association with the Small Business Development Corporation, the Department of Employment and Training and ComputerTown Australia. Further details are available from the conference secretary, Rita Arbuthnott, on (03)781-1777. □

QDP – Fast Z80 Computer

THE QDP, manufactured by Quasar Data Products of Cleveland, USA, is claimed to be "possibly the fastest Z80 computer yet released", but side-steps most of the complexities normally associated with the CP/M and MP/M operating systems.

This makes it 'friendly' to the first-time user, and efficient for the experienced user, according to Dr Simon Rosenbaum, the managing director of Insystems, the QDP's Australian distributor.

The QDP system options include one to four terminals, twin 20 cm 1.2 megabyte floppy disk drives, or a floppy plus 10 or 15 megabyte internal Winchester drive, and both CP/M and MP/M operating systems, with expansion boards for extra RAM and 16-bit processors. It also has a battery-backed real-time clock.

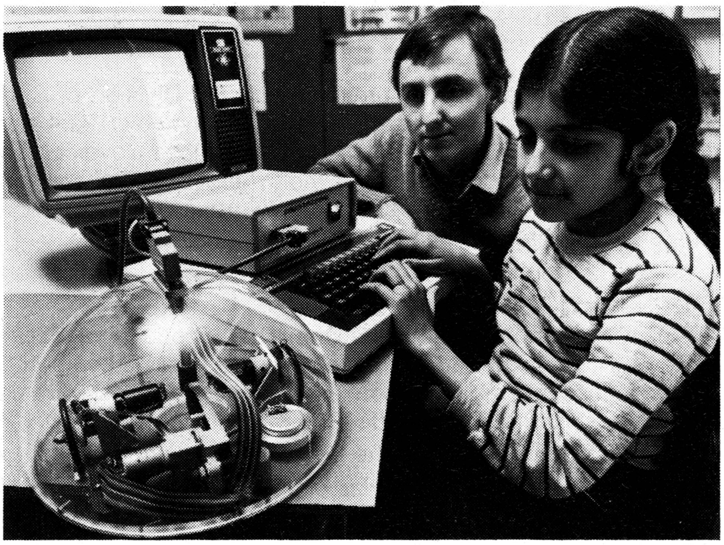
The CP/M QDP comes with its own word processing, spelling, database and financial spreadsheet software, along with a number of utility programs. One of these is SYSTAT, which apart from telling the user how the system is set up, also keeps a record of all disk errors during normal disk operations, thus providing the user with a warning of impending disk or media failure.

The system is fully CP/M and MP/M compatible, allowing it to use most software written for these common systems, but it also contains a 'friendly' menu program to buffer new users from the more complicated aspects of the operating system.

Expansion slots have been left for new boards which will be available later this year. One of these is a high-capacity RAM card which will simulate an extremely fast disk drive, and the other is a 16-bit computer card built on the 80186 chip to the same standards as the existing 8-bit board.

More information is available from Insystems, 337 Moray St, South Melbourne 3205, (03)690-2899, and 84 Pacific Hwy, St Leonards 2065, (02)439-3788. □

The QDP multi-user desktop computer from Insystems.



The 'Turtle' helps teach children plan management and problem solving skills.

The Teaching Turtle

IT'S CALLED TURTLE – a computerised aid to learning that stimulates inventiveness in children and provides them with an entertaining introduction to basic computing concepts. Designed in Britain, the dome-shaped device draws geometric shapes and patterns as it moves across a flat surface in the direction chosen by the operator.

The Turtle is controlled by typing simple commands into the computer from a keyboard. Each key represents a manoeuvre that can be viewed on the visual display as it is being translated into action.

A teacher can write sets of procedures which the child can follow, or the pupil can be allowed to use personal initiative, observing how the Turtle reacts to specific commands and developing skill at drawing and solving problems with a teacher's help. Whatever approach is used, the primary educational value is that a child learns to state plans precisely in a given sequence of commands which are tested by the response of the Turtle. Shapes created can be stored on tape or floppy disk for replay when required.

An Australian Turtle has also been developed by Flexible Systems, 219 Liverpool St, Hobart (002) 34-3064. □

Financial Information Service

THE AUSTRALIAN International Finance Corporation has announced a new information service for corporate treasurers and investors.

Provided through Control Data's Cybertel videotex service, Money View contains a money-market commentary, describing the activity and events in the market, as well as current market yields for a range of securities and information on the major interest indicators in Australia and overseas.

In addition to data on the Australian money markets, the service will include information on local and international stock markets, foreign currency and hedge markets, futures and commodities.

In August, Money View will be expanded to allow subscribers direct access to accounts which they operate with the Australian International Finance Corporation. In addition to account balances and transaction information, they will be able to make or call deposits. This expansion represents the first use in Australia of videotex "gateway" technology, and will directly connect the subscriber to the same nation-wide computer systems that are used by the corporation's money dealers.

The service can be accessed by dialling a local number in any state capital. The subscriber may purchase the colour terminal for $1716 and subscribe for $8 per week, or rent the entire service for $25 per week. Connect time is charged at 20 cents per minute. The unit connects directly to a standard telephone socket and can dial the videotex service automatically. Once connected, the user can scan the database of money market information and store any screens of interest for later recall.

For further information, contact the Australian International Finance Corporation on (03) 602-3233 Melbourne. □

New Educational Publisher

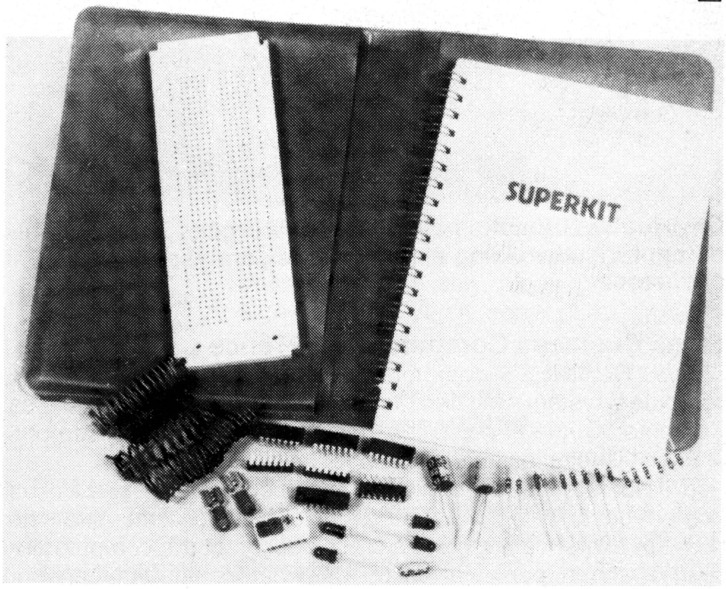
CAMBRIDGE LEARNING, a UK publisher of educational books and kits, has opened an Australian subsidiary to distribute its products.

The company's best-seller is the 'Cambridge Learning Superkit', a practical self-instruction course in digital electronics. It comes with a 'breadboard' on which the circuits can be built without soldering, so the components can be used over and over again. The whole Superkit, including manual, components and breadboard, comes in a pocket-sized wallet and costs $40.50 including postage and packing.

Another self-instruction course available is 'Computer Programming in BASIC'. This course is in four volumes, and claims to teach BASIC in a much simpler way than any computer handbook. It costs $25.80.

'Digital Computer Design' explains the inner workings of a computer, and is priced at $18.50.

These courses and further information can be obtained from Cambridge Learning (Australia), PO Box 173, Kwinana 6167. Ask for a free book list. □



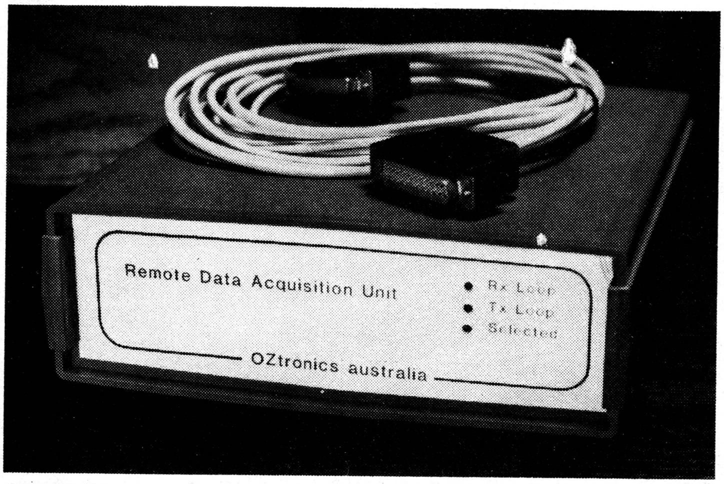
The Cambridge Learning Superkit — a practical self-instruction course in digital electronics.

More Than Just A Pretty Face

IRMA, A NEW board for the IBM-PC, allows the machine to interface to IBM 3270-type controllers such as the 3274 and 3276. IRMA is a decision support interface which permits the user to access and store mainframe data.

The board is supplied with 3278-2, –3, –4 and 3279-2 and –3 emulation programs as standard, and is fully buffered, with a full 1920 character screen page on board. An APL character set is available as an option.

Announcing the release of IRMA, Doug Rattan, the managing director of Sourceware (4/73 Albert Ave, Chatswood, 2067; phone 02 411-5711 Sydney) said more than 4500 had been installed in the United States and several thousand more were on order. □



The OZ1 remote acquisition data unit, which allows almost any computer to be used to gather information from the real world.

OZ1 – A New Data Acquisition Module

OZTRONICS AUSTRALIA Pty Ltd recently released OZ1, a simple, inexpensive yet powerful data acquisition system which allows almost any computer to be used to gather analogue and digital information from the real world.

A supervising computer communicates with OZ1 via the serial port. Consequently, many portable computers, small business computers and low-cost personal computers may be used with OZ1 for data acquisition without modification or addition of special interface cards, requiring only that a serial interface be available.

OZ1 data acquisition units may be connected to the serial port of the host computer either as a single unit, or as a network of up to 16 units in a star or daisy-chained configuration.

Each OZ1 unit provides for 30 single-ended or 15 differential analogue channels and two digital channels, and a fully implemented network provides for 480 single-ended or 240 differential analogue channels and 32 digital channels.

OZ1 has a wide range of applications including real-time data acquisition, process monitoring, automated temperature monitoring, soil and water salinity monitoring, and education.

For further information, or enquiries relating to specific applications of OZ1 data acquisition systems, contact Oztronics Australia Pty Ltd, P.O. Box 375, Belgrave 3160. (03)754-7690. □

Apple Moves Offices

APPLE COMPUTER Australia has moved to North Ryde so that all sales, distribution, service and support functions can be contained in the same offices after the switchover from Electronic Concepts distribution.

For further information contact Apple on (02) 888 5888 or go and see them at 37 Waterloo Road, North Ryde 2113. □

Hard Disk Micro Decision

The Hard Disk Micro Decision is a professional CP/M based computer system that comes with the hardware and software needed for the large or small business. It offers all the word processing, financial planning and programming tools needed to increase personal or business productivity. The hard disk option allows the use of large data bases, increases data transfer speed, and obviates the need to constantly swap diskettes.

Hardware Features

- ★ Z-80A CPU operating at 4MHz
- ★ 64K internal RAM
- ★ Double density floppy disk
- ★ 5 Mbyte (formatted) Winchester
- ★ Two serial ports
- ★ Software programmable baud rates
- ★ Centronics printer port
- ★ Intelligent VDU
- ★ Green screen
- ★ 22 user-programmable function keys
- ★ Detachable keyboard
- ★ Reverse video/dual intensity

Standard Software

- ★ CP/M 2.2 with enhanced BIOS
- ★ Wordstar 3.0 wordprocessor
- ★ Correct-It spelling checker
- ★ Personal Pearl data base manager
- ★ LogiCalc electronic spreadsheet
- ★ Microsoft BASIC 80
- ★ Bazic
- ★ Pilot
- ★ Micro Menus
- ★ Games Disk
- ★ Disk Utilities

Options

- ★ Integrated Accounting System
- ★ Cash Flow System
- ★ Full range of CP/M programs
- ★ Other disk sizes

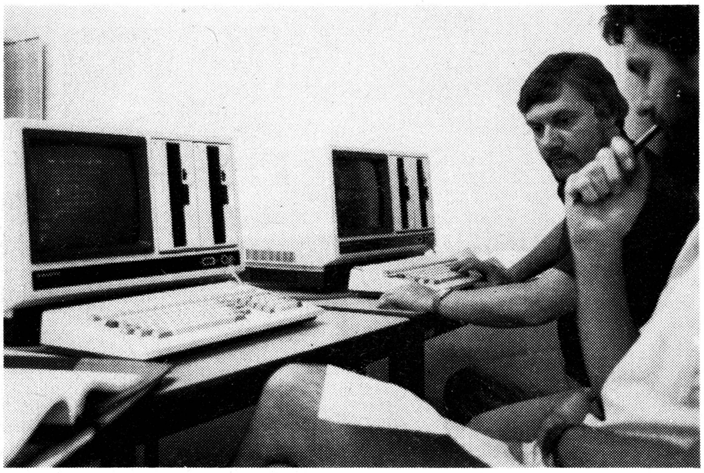
Even though significant hardware enhancements have been made, the special features of the standard Micro Decision such as "virtual drive" and user-friendly error messages have been retained. Disks from other machines such as the Osborne 1, Xerox 820, IBM PC and now Osborne Double Density can be read and written.

The Micro Decision was designed with the first-time computer user in mind. A menu program reduces complex system commands to one or two keystrokes, and a clearly written user manual explains the hardware and software features of the system.

The Hard Disk option was developed in Australia by Microtrix and is only available from recognised Morrow dealers. Upgrades to existing MD2 systems are possible. Microtrix also manufactures a full range of multi-user multi-processor systems. Please contact us for more details.

microtrix

Microtrix Computer Systems Pty. Ltd.
75 Grand Boulevard
Montmorency, Vic., 3094
Phone (03) 439 5257
in Sydney: Automation Statham (02) 709 4144



Woden's TAFE College computing students at work on Sanyo MBC3000 micros.

More TAFE Computer Courses In ACT

THE WODEN TAFE college has responded to a need for more courses on computers and their applications, and has launched several new courses. According to Trevor Farley, head of the computer services department at the college, there is such a great need for courses on microcomputers in Canberra that they have been allowed to conduct courses on the technical aspects as well as on applications.

For the initial courses the college had 250 applications, of which it accepted 130. The courses last for six months, three hours a week, one covering applications of micros and the other programming and basics – technicalities for the novice.

Planned courses include the use of computers, computing studies and accounting, and a BASIC programmers' course.

To give students hands-on experience on micros, the college purchased 16 Sanyo MBC3000 microcomputers, largely because they had the right kind of accounting software, good graphics capabilities, large disk storage and good screen handling facilities.

For more information on the computer courses contact the Woden College of TAFE. □

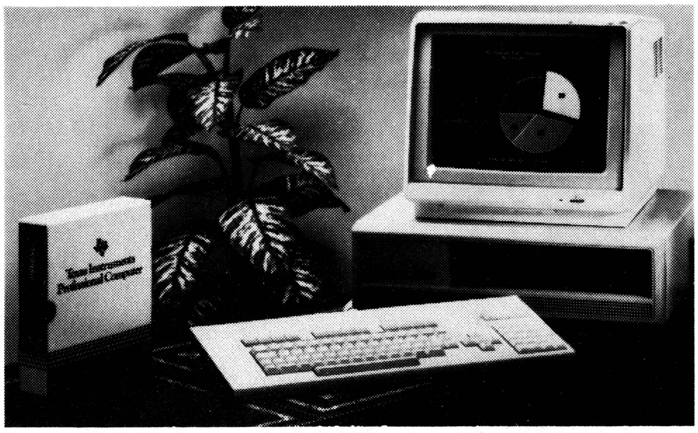
Micros In A Mainframe World

MORE AND MORE microcomputers are making their way onto desks unconnected with the data processing departments of companies. They are perceived as management tools that will solve users' problems without the red tape and time-wasting of the traditional DP set-up.

However, most DP departments and corporate managements don't have a policy on the selection, purchase and implementation of micros, which means the opportunity for maximisation of an asset can be missed through simple ignorance or confusion.

Fletcher DP Services provides corporate-level microcomputer consulting services to advise on the integration of micros into the overall DP facility of a company. Fletcher's consultants have considerable experience in both mainframe and micro environments and are therefore well-qualified to advise on the integration of the two.

Fletcher DP Services is also the developer of specialist turnkey systems for medical, newsagency, publishing and personnel markets. For more details contact Mark Fletcher on (03)527-3443. □



The new Texas Intruments Professional Computer.

TI's New Professional Computer

A PROFESSIONAL desktop computer will be available from Texas Instruments Australia in August.

The TI Professional Computer's capabilities include advanced ease of use and ergonomic features, such as high-resolution colour graphics and a low-profile keyboard; a broad base for the leading industry-standard operating systems and applications software; network communications; and high-system configuration flexibility and expandability.

The TI PC will sell for $4200 for a basic system, consisting of a monochrome display, keyboard, system unit with 128 kilobytes of RAM, and an integral 320-kilobyte floppy disk drive.

The compact system unit contains the 8088 central processor, memory, disk drive and expansion options. Main memory can be expanded to 256 kilobytes. Space is provided for a second floppy drive or a five- or 10-megabyte Winchester disk.

The TI PC uses a 30 cm monochrome display or an optional 33 cm colour display. Both displays use the same format – 25 lines of 80 columns, and 720 by 300 pixels with the graphics controller option.

A variety of communications options, including TTY and 3780 emulators, is available for the TI PC in network environments.

Earlier this year, Texas Instruments introduced a new low-cost impact printer, the Omni 800 Model 850, as a companion printer to the new TI PC computer. The Model 850 prints at up to 150 characters per second and offers a variety of fonts and print options. The Omni 850 also features a "raster graphics" capability which can be used to print graphics designs from the TI PC display.

The TI PC supports four of the leading 16-bit industry standard operating systems: MS-DOS, CP/M-86, Concurrent CP/M-86 and UCSD P-system.

A variety of accounting, financial modelling and planning, database access and management, graphics and word-processing packages are also available.

For further information, contact Texas Instruments Australia, 6-10 Talavera Rd, North Ryde, 2113. Phone (02) 887-1122. □

Datronics Handles Harris

DATRONICS GRAPHICS Systems has announced that it has signed a distributor agreement with Harris Information Systems International Division.

Datronics will represent Harris in Australia to market its range of distributed data-processing, interactive terminals and remote batch entry systems. □

THE DATABASE HOTLINE!

THE dBASE II EXPERTS

★ THE HOTTEST line in database management — Ashton-Tate's dBase II. The most popular DBMS on the world micro scene, it's one of the all-time record sellers. And for good reason.
★ dBASE II is a complex and extremely powerful system, yet even the novice can use it to control vast data structures.
★ A HANDFUL of straightforward English commands — like CREATE, EDIT, APPEND, or INSERT — is all you need to know to start using dBase II right now.
★ A POWERFUL English-like command language gives you almost unlimited control of your computer and its files. Full scale accounting systems have been written — by businessmen — using this language and dBase II's database management alone.
★ NEW: Version 2.4, just released, features several major enhancements.
★ WE SUPPORT dBase fully. We use it ourselves, for just about everything! If you want to know how to get something done with dBase, we're the people to talk to.

★ FRIDAY! — Son of dBase II!

★ The database management system from the masters, Ashton-Tate, for those who want real power without the need for programming.
★ HIGH POWER, low cost: around half price of dBase II.

Get it all done by Friday!

★ Friday is easy enough to use at home, powerful enough for the office.
★ Set up an "electronic file" without even peeking at the manual.
★ Find any filed information in seconds, even if there are thousands of entries in the file.
★ Sort files on up to 5 different items.
★ Change the structure of a file with a few keystrokes, without losing any data.
★ Display or print quick reports from all or part of a file.
★ Prepare custom reports laid out just the way you want them, quickly and easily.
★ Set up your system to perform automatic calculations when data is being entered or when reports are being printed.
★ Protect your files with passwords.
★ Easily merge mailing lists with form letters using Friday and a word processor.
★ Make handy mailing labels any way you want them.

Friday is worth the price just for the way it handles mailing lists and labels!

archive

Computer Services Brisbane

P.O. Box 13, 23 Wagner Road, Clayfield, Queensland, 4011 Phone (07)262 2911 Telex 44187 ARCHIV

Office Automation Seminars

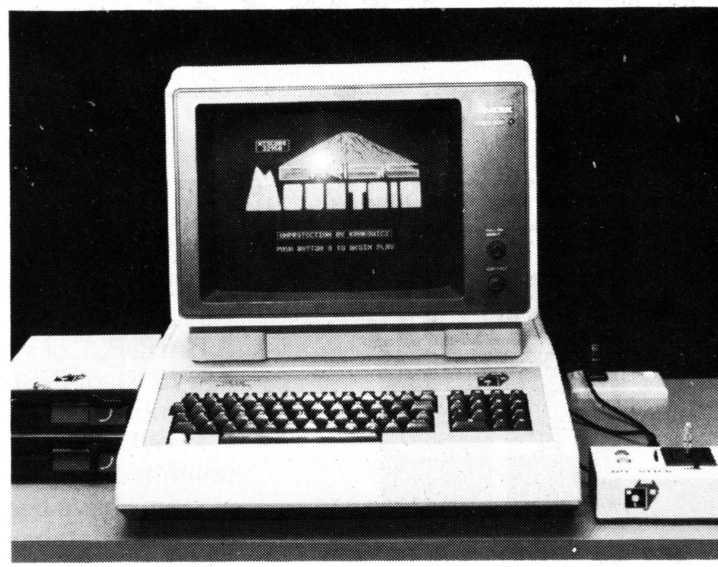
STARTING IN AUGUST, Wang Computer will be running seminars to take managers through the concepts of office automation. Each session is constructed to take twelve participants, will last two days, and is open to individuals or company bookings.

The seminars are scheduled as follows: Sydney – August 17-18; Canberra – Sept 8-9; Melbourne – Sept 12-13; Adelaide – Sept 15-16; Perth – Sept 19-20; Brisbane – Sept 26-27.

Leading the seminar program will be Elaine Ray, co-author of two publications which have been among the largest-selling software and hardware computer books on the Australian market over the last three years.

The seminars are divided into three major segments: office functions and automation benefits; strategic approach to office automation; implementation. Participants will also be introduced to the concept of six technologies being involved in office automation: data processing, word processing, image processing, audio processing, networking and human factors.

For further information contact John Baker at Wang on (02)436-3477.



The Australian-built Fox-640 multi-system computer, claimed to be compatible with most available systems.

Fox Multi-System Computer

FOX COMPUTERS' new Fox-640 multi-system computer has a 64-kilobyte memory board with eight expansion slots, but has no ROM and no fixable language.

The Fox-640 is said to be compatible with the IMC OS card, IBM's AD8088 CPU ALF card, MicroSoft, CP/M, the Z80 softcard, Apple's 80-column card, Integer card, 6809 card and the RS232 disk controller card.

The operating system, FOX DOS, is written in Australia and the Fox-640 is Australian-built.

Standard features of the Fox-640 include a Forth programming system card, joystick port, 13 pre-programmed function keys, cassette interface, numeric keypad, two speakers, a volume controller and a blank system card on which you can create your own system.

Recommended retail price of the Fox-640 is $980, plus tax.

More information can be obtained from Fox Computers, PO Box 203, Hornsby, 2077. Phone (02) 476-4582. □

File Transmission First

A SYDNEY software house has developed an all-Australian file transmission system that is set to capture a major part of the local and overseas markets.

The group, Software Developments, claims a world first for the file transmission services (FTS) breakthrough.

"To our knowledge, it's unique in current product offerings world-wide," said manager John Robinson.

The FTS option will allow the transmission of sequential files and members of partitioned data sets from one computer site to another. This will result in faster data movement and enormous cost savings for users who have moved information by manual methods in the past.

"Moving disks and tape by courier from state to state will be a thing of the past for our FTS users," said Robinson. "And there will be further savings on time lost when tapes or disks are in transit." □

CITP Program Expanded

THE COMPUTER Industry Training Program (CITP), which consists of four weeks' formal COBOL training and four weeks in a workshop environment, has added to its curriculum

another two segments covering RPGII and III and BASIC.

The CITP is financed by the Federal Government, while the New South Wales State Government provides the machinery for trainee selection, training through TAFE and administration. The scheme is one of the Federal Government's "skills in demand" programs, aimed at alleviating unemployment and assisting the computer industry by supplying skilled programmers.

Employers who take on CITP trainees are eligible for a subsidy of $100 a week during formal training and $86.90 during the first six months on the job.

For more information, contact Ormonde Brown at the Australian Computer Society, phone (02) 290-3887. □

New Ada Compiler

DATA GENERAL HAS announced what it claims to be the first complete range of Ada development tools to be released by a major computer vendor. Previous Ada compilers have generally been released by microcomputer companies.

Ada is the latest attempt by the United States Defence Department to design a completely standardised programming language which can be used for all applications within the department.

The Data General Ada, which runs on the MV family of Eclipse computers, is offered by arrangement with the Rolm Corporation, and is a full ANSI MIL-STD 1815A-1983 implementation, which has been extensively tested using the Defence Department's validation test suite.

Data General has packaged up the compiler, along with various development tools and the necessary hardware, and hopes to capture a larger share of the United States local defence market. □

Work Skills Program in Business Communications

A SPECIAL work skills program is currently in progress at the Seaforth and Sydney TAFE colleges to provide unemployed young people with basic career skills in data input and word processing. This is a special Transition Education course to help such people to find work in the business communications industry.

Employers are finding that young people who have completed Transition Education courses are better equipped than the average unemployed unskilled person looking for a job.

Apart from giving students a clear understanding of the basic skills required for employment in the business communications industry, the courses aim to develop self-esteem, social skills, job-seeking skills and survival skills.

Work Skills courses are geared to the employment needs of the 15-19 age group, and the courses involve full daytime participation for from 12 to 18 weeks. People participating get a Transition Allowance equivalent to the unemployment benefit plus $6 per week, as well as travel and health benefit concessions, a $30 book allowance and a living away from home allowance.

Intensive practical training in a particular career option is augmented with work experience days arranged with local employers and visits to factories and offices. Course co-ordinators co-operate closely with industry employers to arrange follow-up employment interviews for graduating students.

The success rate of Work Skills courses is high; some young people participating find employment immediately on completion of the course, and others are offered jobs during their training period.

Unemployed young people wanting to know more about Work Skills courses shou J contact their local CES office or TAFE college. □

Nationwide Real Estate Information Network

AUSTRALIA WILL soon have a nationwide property listing and referral service, REINET, introduced by the Real Estate Institute of Australia.

According to the Institute, this new marketing tool will enable subscribers to transmit and receive information nationally and internationally on industrial, commercial and rural properties for sale, lease or exchange, hotels, motels, multi-unit dwelling blocks and prestige residential properties.

REINET was first launched in the USA and now has over 600 subscribers; a European service began in September, and talks for Asian services are being held with Singapore and Japan. The Australian service is expected to begin in August.

The REIA has also just published a revised and updated edition of 'Small Computers for Real Estate Offices' to coincide with the launch of REINET. This publication surveys the availability and range of micros, and aims to inform small amd medium-sized estate agencies of low-cost computer facilities relevant to their operations.

A feature of the new edition is an evaluation and check list for agents considering computers for the first time, and there is another new section on computer terminology. Twenty-seven systems are described in the $5000-$25,000 price range, and all are suitable for hooking into the REINET service.

'Small Computers for Real Estate Offices' costs $12 and is available only to Institute members. For more information on both it and REINET, contact REIA, Canberra Savings Centre, City Walk, Canberra. (062)47-5533. □

Small Business Accounting Systems

A COMMON complaint about most small business software is that it is too pretentious and is just a chopped-down version of someone's big business package.

Ilehead Pty Ltd is exclusive distributor of the CISA Small Business Accounting System, which it claims will fill the need for an accurate and fully integrated system which is user friendly and not too pompous and time-consuming.

The CISA system was designed and written in Australia and aimed specifically at the typical small business, and no previous accounting or record-keeping experience is assumed. Data entry is said to be virtually foolproof, basic information being entered into the computer from cheque book stubs, invoices received, bank deposit books and bank statements. Entries can be made at any time and in any date order.

The system is based on the 'perpetual accounting' principle, in that no entry is ever erased or updated. For example, a full statement for a creditor can be instantly printed, showing all transactions right back to the beginning of the financial year, including the balance carried forward from the previous year. The system is in fact a fully automatic double-entry general ledger system, including a purchases (or creditors) ledger.

The system is driven from a single menu, and the documentation is said to be easy to read and contain lots of information on interpreting financial statements and saving time in record-keeping.

As a special introductory offer Ilehead is offering $100 discount on the twin-disk SBAS3 system, usually priced at $350. The program is currently only available on Tandy Models I and III and the System 80, but preparations are said to be advanced to put the software onto other popular 8-bit and 16-bit systems.

For further information contact Ilehead on (02)412-3470. □

All The News That Fits

THE INTEL CORPORATION has announced an agreement with Western Electric to port Unix System V on to the iAPX 286 microprocessor. Intel will now support a wide range of operating systems, including Xenix, iRMX, MS-DOS, CP/M and its original ISIS.

Leading market research company Roy Morgan Research Centre has ordered a new data-entry system from AWA. The system, based on CMC-Pertec XL 40 equipment worth around $100,000, will be used by Roy Morgan subsidiary Audits and Panels to analyse information from 3000 households on multi-brand buying, brand switching and other issues.

A new plotter from Nicolet Zeta is designed to plug directly into IBM 3274 or 3276 cluster controllers. The eight-pen plotter supports a variety of media, and can be driven by standard software such as DISSPLA, Tell-a-graf, DI3000 and SAS/GRAPH. The local agent is EAI-Electronic Associates, phone (02) 439-7522 Sydney.

The Standards Association of Australia has published a revised standard for FORTRAN. AS1486 describes both FORTRAN and Subset FORTRAN as laid down in the ANSI standard X3.9-1978, and better known as FORTRAN 77.

Mitsui Computer Systems, the agent for Sord, Diablo, Fujitsu, Hazeltine and others, has opened a new sales and service office at 303 Coronation Drive, Milton, 4064. The phone number is (07) 369-7799 Brisbane.

Datapoint has announced a new cut-sheet feeder for its model 9611 35 characters per second letter-quality printer.

Hewlett Packard has released a new interactive graphics tablet which allows users of the 2627A colour graphics terminal to create their own drawings and sketches.

Occidental Life of Australia has upgraded from an IBM 4331 Group II central processing unit to a 4341 Model LII, in a changeover worth more than $500,000. The replacement will be made two and a half years earlier than originally planned.

Sanyo Business Machines has announced a typesetting program for its MBC-1000 microcomputer.

Honeywell Australasia has signed an agreement with a Paris software house, Steria, for the French company's videotex software package, Videopac, which runs the French Teletel service. Under the agreement, Honeywell will market the package in Australia, New Zealand, Papua New Guinea and Fiji.

DD WEBSTER ELECTRONICS has announced a new spreadsheet calculator and financial modelling package for its Spectrum-11 computers. Called Saturn-Calc, it will also run on PDP-11 and VAX machines. It is priced at $500 for a single-user version and $1000 for a multi-user licence.

TEKTRONIX has announced Plot-10 Easy Graphing II, a low-cost graphing command language that is compatible with all Tektronix storage and raster terminals. A Fortran version is available for most mainframes at $4000, while a compiled version is available at $500 per copy to run on Tektronix 400 series terminals under CP/M-86. Tektronix has also announced a 10-megabyte hard disk for the 4100 series.

THE NEW LOW-COST eight-kilobyte computer keyboards promised earlier this year for the Atari 2600 games system have been postponed, according to Futuretronics, Atari's distributor in Australia. Heavy commitments to production of NTSC keyboards for the United States market have meant that production of PAL keyboards – such as those for the Australian market – has been postponed till 1984.

DATASCAPE, THE printer specialist which started operating in Sydney 16 months ago, has now opened an office in Melbourne at 27 Raglan St, South Melbourne,phone (03) 690-3622. Datascape is the Australian representative for the Anadex Silent Scribe and Word Scribe range, the NDK S7700 and the family of NEC Spinwriters. The Melbourne office will also support Datascape's Instrumentation Division.

ALL MICROCOMPUTER software manufactured and distributed in Australia by Imagineering now has a full Australian warranty. Major American companies such as Visicorp have supported Imagineering in introducing this facility, which will mean that products can be repaired or replaced promptly without having to be sent back to the United States.

DIGITAL ASSOCIATES has released two new IBM-compatible printers, manufactured by Hitachi. They print at 1250 and 2000 lines per minute, and are modern high-speed impact band printers using a 48-character set. Controlled by microprocessors, they connect directly to the mainframe channel through IBM's bus and tag cables, and installation takes only around two hours. The printers will retail for $45,000 and $65,000 respectively.

COMPUTERMAT, at 665 Pittwater Road, Dee Why, 2099, phone (02) 982-3288, has started training courses for first-time users of computers. A10-week BASIC programming course starts in August, and other courses are being organised for business people, potential programmers and secretaries.

EVERYONE HAS heard of the little cursor-moving device known as a mouse – well, now you can buy one that's compatible with the IBM-PC or PC-compatible computers. It's the Logimouse, a Swiss-made rodent that has undergone three years of tests in various technical and design environments, and which requires no special software. For more information, contact Microhouse, PO Box 642, Unley, 5061, phone (08) 272-4370. □

Oooops! – Tandy Teaches Too

IN OUR MAY issue article surveying computer educational courses we neglected to mention Tandy, which pointed out to us somewhat irately that it has been running regular courses around Australia for several years.

These courses include programming training, applications training and teacher training, and some specific courses for teachers and school administrators are also regularly scheduled. Since last year Tandy has also run vacation computer schools for schoolchildren, which have proved popular.

Each Tandy training centre has rooms fully equipped with 15 microcomputers and so provides valuable hands-on experience.

For further information on Tandy's computer courses, contact Tandy's Education Division, at the company's new head office and warehouse complex at 91 Kurrajong Ave, Mt Druitt 2770. (02) 675-1222. □

Home Finance Software For MicroBee

ALLSOFT COMPUTER SERVICES has designed a home budgeting software package for the MicroBee, designed to fill the gap in this area in currently available MicroBee software.

The package consists of two independent programs, BACCS for home accounting and BPLAN for budget planning. The programs use forms and techniques with which most people are already familiar, and operate in a user friendly manner.

The package has a recommended retail price of $28.90, and is not intended to be sold through mail order. Intending purchasers should contact MicroBee software dealers.

Dealer enquiries are welcome, and should be made to Allsoft Computer Services, PO Box 78, Charnwood ACT 2615. (062)58-6864. □

Computerised Navigation

THE MELBOURNE-based America's Cup contender, Challenge 12, is using a completely new, locally developed computerised performance measurement and navigation system in its quest for yachting's most illustrious trophy.

Developed by Damar Management Systems, of Geelong, the Challenge 12 system is running on computers and peripherals provided by Hewlett-Packard Australia.

The systems were given their real test during the Westpac Series sailed on Port Phillip Bay during March, and the crew of Challenge 12 is said to have felt its electronic performance analysis and navigation capabilities gave the boat a potentially significant edge.

Further information can be obtained by contacting Damar Management Systems on (052) 22-4322. □

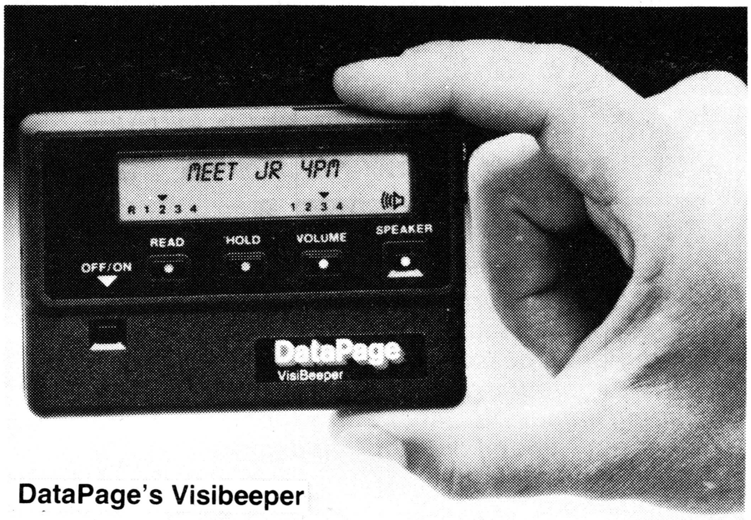
Pitman's Enters Computer Education Market

PITMAN PUBLISHING has announced the formation of a new division, Pitman Education Software (PES), to cater to the growing education market for Australian and imported software.

PES has just released its first catalogue of software for the BBC microcomputer, and intends to supply software for other micros, including the Apple II Plus.

The company's first undertaking in the computer education market has been in association with the BBC to introduce the BBC Computer Literacy Project to Australia. The project includes the BBC microcomputer, software, and ten half-hour video programs currently screening on ABC schools television.

For further information contact Mariel Beros, Publicity Manager, on (03)347-3055. □



DataPage's Visibeeper

No More Beeps

A NEW KIND of pocket pager has just been released in Australia – the Visibeeper from DataPage. Instead of beeping to attract the user's attention, the Visibeeper displays the message in words and numbers on a LCD screen.

Text messages of up to 80 characters can be received directly by the pager, and as many as four messages totalling 160 characters may be stored in the pager's memory. At the user's command, the message moves across the readout screen at a comfortable rate, and the readout may be stopped and held at any time.

At present, you can only operate a Visibeeper in Sydney or Melbourne, but the service will be extended soon to other centres, according to DataPage, which recently installed a $1 million computer and radio transmitting network to allow the instrument to be used within factories, office complexes and building sites.

For more information on the Visibeeper or DataPage's other services, phone (02) 923-1122.

NEC also offers a display pager, the R1D3-1B low-band or R3D3-1B high-band, which can store up to 40 characters, comprising numbers and the letters a, b, c and d. A new pager now being developed by NEC will have both numerals and the complete alphabet.

For more information, contact NEC on (02) 438-3544. □

IBM Supports Austpac

IBM HAS formally announced support for the attachment of IBM communications products to Telecom's Austpac Packet-Switched Network (see Networks article in this issue's *Your Business Computer* supplement).

The announcement follows extensive testing in Australia of the previously announced IBM X.25 communications support for products such as visual display units, communicating Personal Computers and the Displaywriter. □

Electronic Games Exhibition

ELECTRONIC GAMES and Toys '83 will be held in the Sydney Entertainment Centre from 18-21 August, and the organisers expect at least 50,000 visitors to attend.

This is the first exhibition specialising in electronic games and toys, and is designed to help the manufacturer, wholesaler, supplier and retailer to reach their appropriate markets. The first two mornings of the exhibition have been set aside for the trade, but otherwise the show will be open to the public until 9 o'clock each evening.

Visitors will not only be able to browse and choose, but also in some cases to order or buy games, as arrangements have been made for some retailers to sell on the spot.

For more information contact the organiser, Industrial Presentations Australia, 4/389 Victoria Ave, Chatswood 2067. (02)412-4377. □

August Seminars

VARIOUS SEMINARS of interest to computer professionals and management will be taking place in August in Sydney, Melbourne and Brisbane.

Management Technology Education in conjunction with Monadnock International is presenting two related seminars, the first on data communications network design and the second covering local area networks (see article in *Your Business Computer*, page 6).

'Data Communications Network Design' will cover the types of communications systems available, data communications codes, formats and error-checking techniques, the advantages of the different network architectures, network protocols, the concepts and features of network subsystems, selection considerations for data communications hardware and software, and network planning.

'Local Area Networks' will focus on the practical realities of LANs, and cover LAN characteristics, their role in network design, selection consideration for LAN communications, the major commercial LAN systems, diagnostic and monitoring design, and network installation and management.

'Data Communications Network Design' will be held in Melbourne from August 1-3 and in Sydney from August 8-10, while 'Local Area Networks' is in Melbourne from August 4-5 and in Sydney from August 11-12. Two more seminars from MTE are 'Fourth Generation Languages' and 'Corporate Communications', both to be held in Sydney and Melbourne.

'Fourth Generation Languages' aims to "objectively examine the claims and performance of fourth-generation language products".

'Corporate Communications' will examine communications from the level of in-house through to interstate site communications, and is aimed at communications managers, office equipment managers, EDP managers, planning managers advising management, and senior executives with a role in company communications policy.

'Fourth Generation Languages' will be held in Sydney from August 15-16 and in Melbourne from August 18-19, while 'Corporate Communications' is in Melbourne from August 15-16 and in Sydney from August 22-23.

Contact Management Technology Education, GPO Box 3939, Sydney 2001, for more details or phone (02)29-3030.

Another August seminar is 'Detection and Prevention of Computer Crime', presented by Drake Data and Drake Accounting, two operating divisions of Drake Personnel.

Computer crime is a nebulous subject and therefore often tacitly ignored by many companies using computers. This seminar will cover typical blind spots in security, management ignorance or abdication of responsibility, system design flaws, how the Australian legal system handles this new crime, and other relevant topics.

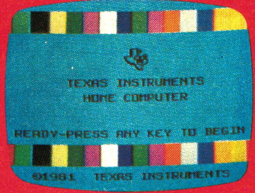
The seminar will be held in Brisbane from August 4-5 and in Melbourne from August 11-12. For more information contact Len Meyer or Tom Smiley on (03)654-4855 for Melbourne bookings, or Barry Ballance on (07)221-6099 for Brisbane bookings. □

TEXAS INSTRUMENTS COLOUR KOMPUTER

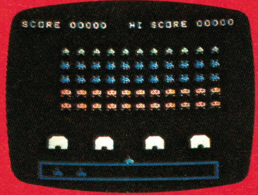
ACTUAL SCREEN PHOTOGRAPHS NOT ARTIST'S IMPRESSION

PARSEC

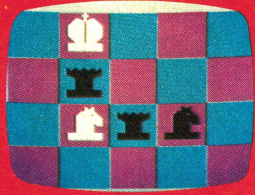
LEARNING



FULL COLOUR



TI INVADERS



CHESS

✓ WHY GET IT?

TI-99/4A

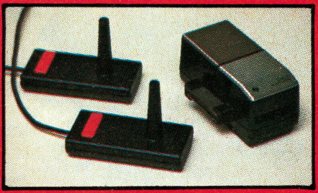
- ✓ FAST 16-BIT PROCESSOR
- ✓ AMAZING HIRES COLOUR GRAPHICS
- ✓ BRILLIANT SOUND EFFECTS — 3 CHANNEL SOUND
- ✓ FAST, FAST GAMES WITH COLOUR AND SOUND
- ✓ JOYSTICKS/SPEECH SYNTHESISER. WOW!
- ✓ FULLY PROGRAMMABLE. VERY POWERFUL
- ✓ EXCEPTIONAL EDUCATIONAL SOFTWARE
- ✓ FANTASTIC SOFTWARE SUPPORT

FULLY EXPANDABLE!

- ✓ PRINTER, DISKS, REMOTE COMMUNICATION
- ✓ ASSEMBLER, TERMINAL EMULATOR, BASIC
- ✓ TI-LOGO 11, EXTENDED BASIC, PASCAL
- ✓ POWERFUL WORD PROCESSOR OPTION
- ✓ ALL BUSINESS APPLICATIONS
- ✓ AFFORDABLE SPEECH SYNTHESISER
- ✓ FITS YOUR COLOUR TV SET
- ✓ AMAZING VALUE FOR MONEY

ONLY $499

SEE IT, HEAR IT, EXPERIENCE IT,
AT THE FOLLOWING QUALITY COMPUTER STORES



OPTIONAL SPEECH SYNTHESISER AND JOYSTICKS

PSST! 'BUSINESS WEEK' FEB. 83 (USA) REPORTS THE TI-99/4A IS SELLING AT A **"RED HOT"** 30,000 UNITS **A WEEK!**

THE COMPUTER

✓ GET IT RIGHT ✓ GET IT FROM ✓ GET IT NOW

COMPUTER DYNASTY
15 FLORENCE ST, HORNSBY
NSW 2077 (foot of rail overpass)
Telephone: (02) 477 6886

Aussie HOME COMPUTERS
SHOP 7, SOUTHLAND SHOPPING
CENTRE, MAWSON, ACT 2607
Telephone: (062) 86 2277

GREYTELLA COMPUTERS
315 CLOVELLY RD, CLOVELLY
NSW 2031
Telephone: (02) 665 9569

OR GET IT BY MAIL ORDER FROM SOFTWARE CONNECTIONS, P.O. BOX 375 PYMBLE, NSW 2073

✓ TICK		CHEQUE		MONEY ORDER		BANKCARD WELCOME	
PLEASE SEND ME	QTY			QTY			QTY
TI-99/4A COLOUR COMPUTER		499.00	TEACH YOURSELF BASIC		34.95	VIDEO CHESS	69.95
SPEECH SYNTHESISER		149.00	PARSEC		49.95	**FREE CATALOGUE**	
PAIR JOYSTICKS		39.50	TI INVADERS		39.95	**FREE EXPANSION PRICE LIST**	
			ADD $9.00 P + D (+ 10% OVERSEAS)			TOTAL AMOUNT PAYABLE	

BANKCARD No.__________ Exp. Date —/—/— SIGNATURE __________

NAME __________ STREET __________ CITY __________ POSTCODE __________

Cash & Carry Computers

– or, The Slashers Strike Again!

As prices drop in the home computer market, the competition is hotting up. Here Les Bell looks at the latest crop of under-$500 machines to see what's what ...

TWO YEARS AGO, we ran a story on 'Slashing the Cost of Home Computing', as we were amazed at the price reductions and new low-cost computers that were appearing. Well, we're still amazed. It is hard to believe that you can get so much real computing power for so little money!

Of course, at these low prices, you don't get – for example – the mass storage facilities afforded by disk drives. By and large, your mass storage takes the form of cassettes. However, disk drive controller cards are starting to appear as an option on an increasing number of small home computers.

Virtually all these machines have colour displays and will accept plug-in games cartridges, so they will do double duty as educators and entertainers. When the kids get bored with creating coloured shapes, they can always blast away at some for a while!

Another new feature that has started to appear only recently is the built-in joystick, mounted next to the keyboard. This often doubles as a cursor controller. I doubt that it will be long before low-cost home computers are supplied with a mouse!

Anyway, read on, and compare these little tykes with your $10,000 S-100 boat anchors. Eat your heart out, Altair!

Microbee IC

If you've been in Australia for a while and haven't heard of the Microbee, you haven't been reading the papers or magazines or watching television. The Microbee was launched on an unsuspecting world in the February 1982 issue of *Your Computer*, and proved to be an enormous success. Initially available in kit form at just under $400, it has since been upgraded and improved, repackaged and generally changed into a bigger computer than it used to be.

The Microbee IC is the latest incarnation of the little mite, and offers the most popular enhancements and options, together with a few improvements, all in one package. The IC is faster (3.375 MHz clock) than earlier models, and incorporates as standard both the WordBee word processor ROM and the NETWORK communications ROM.

The IC uses MicroWorld Colour BASIC V5.22, which includes additional commands to set the foreground and background colours and modes. Thirty-two colours are available for the foreground, not all of them describable, and eight for the background.

Listings can now be set to be in either upper or lower case, according to the user's preference; typically, I find lower case easier to read.

It seems that Applied Technology is planning to release more software in ROM form for the Microbee. Up to (theoretically) 256 different ROMs can be plugged in, and the command *PAK n* will select the appropriate ROM pack by outputting the value of *n* to the memory bank select port.

In the IC, two ROMs are provided as standard. Most useful probably is the WordBee version 1.2 ROM. WordBee is loosely modelled on WordStar and Electric Pencil, and incorporates a surprising number of useful and powerful commands for such a small system. Version 1.2 contains several new features, such as the ability to vector output to one of a number of outputs, which gets round a major problem for many Microbee owners. In addition, touch typists can select input from an external keyboard which they may prefer.

Other new WordBee commands include underlining and double striking, and a new command allows the user to move the cursor to the end of the current line.

The other major addition is the terminal/network ROM which not only provides communications facilities, but provides a number of other general tricks accessible from BASIC or elsewhere. The general NET command will turn the Microbee into a full or half duplex terminal with an 80 by 24 screen which emulates most of the codes of the Televideo 912 terminal.

The baud rate is settable at 110 to 4800 baud, and parity can be odd, even or off. Best of all (to us here at YC particularly) the NET ROM implements file transfer using the Christensen protocols, so that Microbees can now communicate with each other and the popular bulletin boards.

The network ROM is accessible from BASIC or from within WordBee, providing a range of extra communications and screen formatting options.

The documentation for the Microbee is continually improving, and the latest versions of the user manuals are very good indeed. The Microworld BASIC manual is well organised for both tutorial and reference use, and is quite readable.

The Microbee has always been a powerful and capable little computer, but this latest version really is a winner. Its design is oriented towards useful activities, such as word processing and communications, rather than game playing – but a heap of games are available if you want them!

Dick Smith VZ200

At just under $200, Dickie's come up with another winner here. The VZ200 is a neat little computer indeed.

The VZ-200 is virtually a totally non-technical machine for the user who wants a gentle introduction to BASIC programming and home computing. For example, nowhere in the manual does it say what kind of processor is under the hood! Indeed, there is virtually no technical detail at all anywhere in the manual.

All this is possibly to the benefit of the completely non-technical novice who could do without that kind of intimidation. But it bodes ill for the future availability of professionally written games and utility software. I'd say that for the near future at least, and excluding whatever Dick Smith may release, the VZ200 will remain a BASIC-only machine.

The VZ200 is probably based on the ubiquitous Z-80, and is supplied with 8 Kbytes of RAM as standard. A 16 Kbyte memory expansion module is available for $79.

The BASIC interpreter used is, of course, Microsoft's Extended BASIC, complete with colour graphics and sound commands. The screen displays 16 lines of 32 characters each, and the keyboard is a calculator-style QWERTY with a soft action. Like most of the machines covered, the spacing between keys was less than I would have liked; obviously they are designed for somewhat smaller fingers than mine.

Two graphics modes are available: in mode 0, the graphics resolution is 64 by 32 pixels with nine colours available and text displayable. In mode 1, the resolution is 128 by 64 pixels in eight colours, and this is a better mode for games and more complex graphics.

The graphics statements are the standard kind used in the TRS-80 Colour Computer and other machines with Microsoft Colour BASIC. A point is set with the statement SET (X,Y) and turned off with the RESET (X,Y) statement. POINT(X,Y) will return true if a point has been set and false if it has not. The colour is set using the COLOR statement, which sets the foreground and background colours.

The background can be either green or orange in mode 0, while in mode 1 only four colours can be selected for each background colour.

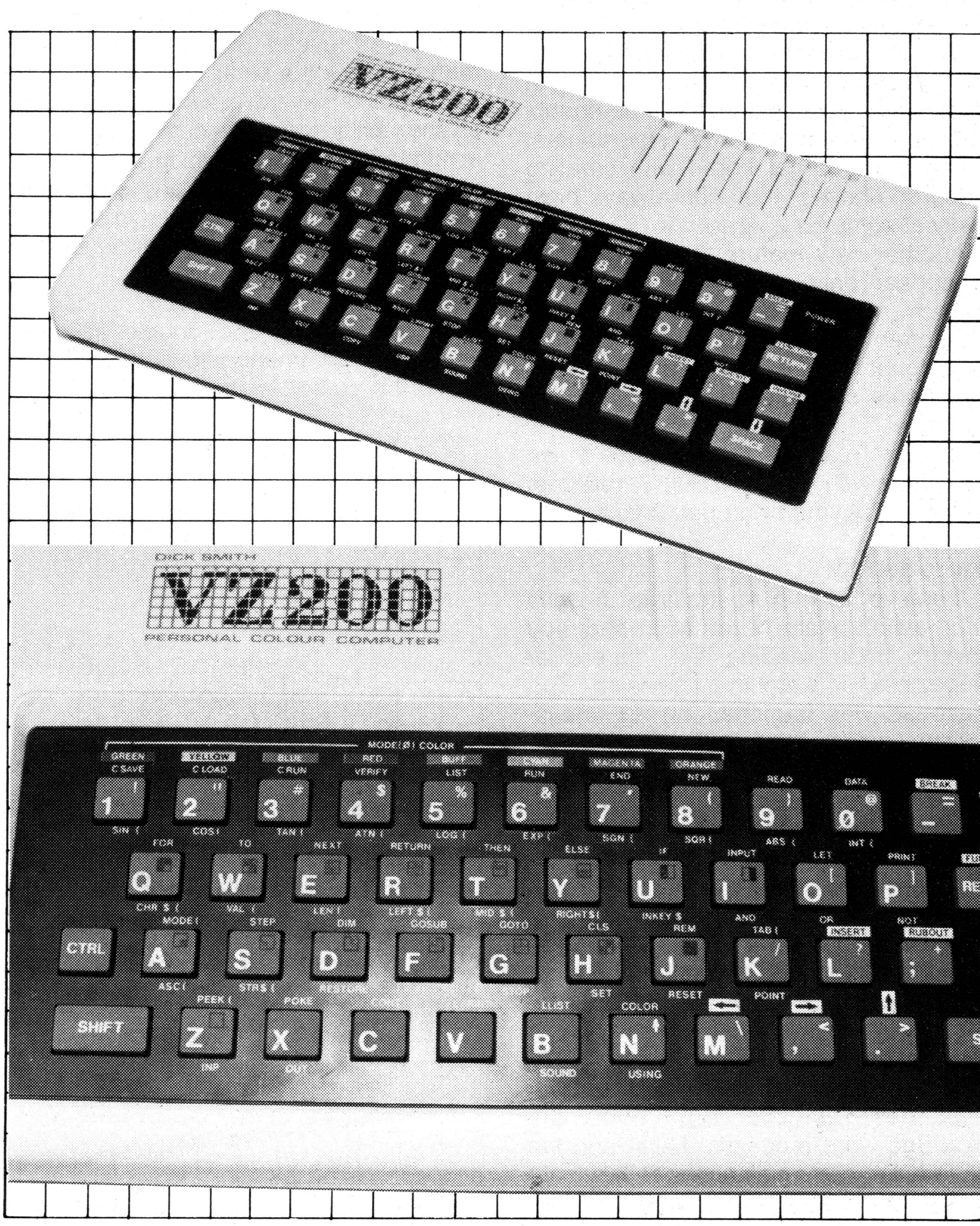
The SOUND X,Y statement will generate a tone of pitch X and duration Y. By using data statements, it is possible to create quite complex little tunes.

For those who want to dabble in some PEEKing and POKing, the manual does give the addresses of the screen RAM, so some fast updating can be done that way, though this will require some experimenting.

The manual is well written, and is organised as a tutorial text, bearing in mind the likely audience for this kind of machine. There are no signs of the Janglish that usually mars manuals on this kind of machine.

Expansion is limited on the VZ-200; there is a socket on the back for the plug-in 16K RAM module, and a peripheral connector, obviously intended for a printer. Apart from the cassette cables, that's it. For parents who don't want their kids to get carried away buying more and more extras, that's probably a blessing!

While this computer probably won't do much for the dedicated enthusiast who wants to get into machine code programming and interfacing all kinds of peripherals, it's just right for those who want to learn some programming and not get bogged down in unnecessary details. Run a business it won't; draw you in to the joys of programming it will!



Spectravideo SV-318

This little package starts off with quite a small personal computer, but it is expandable into a full computing system of quite useful proportions.

The basic console, which contains the computer proper, is only a few inches deep and not that much bigger than, say, the Sinclair machines – but it includes 32 Kbytes of ROM and 32 Kbytes of RAM as standard. The keyboard is a calculator type with a soft feel but a reassuring amount of travel, and is easy to use.

The keyboard includes all the standard QWERTY characters, including the tilde and the escape and control keys, but it also includes five shiftable function keys (which are pre-programmed for BASIC keywords) and some miscellaneous keys for functions such as character insertion and deletion. In addition, the 71-key keyboard is marked with a set of graphics characters.

An unusual feature of the keyboard is the joystick/cursor control pad at the right, which can either be used as a cursor pad for editing, or with the joystick plugged in to double as a games controller.

Inside the box there's a Z-80A microprocessor running at 3.6 MHz, with 32 Kbytes of ROM containing Microsoft BASIC and 32 Kbytes of RAM, half of which is dedicated to graphics. The ROM is expandable to 96 Kbytes using plug-in cartridges, while the RAM can be extended to 256 Kbytes.

The graphics capability of the SV-318 is impressive. The screen resolution is 256 by 192 pixels, with 16 colours available. Most important to games creators, however, are the 32 sprites which are available. These movable shapes can collide with each other and other objects, or can pass in front of or behind each other.

The SV-318 also provides three sound channels, fully controlled by the built-in BASIC, which will allow the user to write music or provide background sound effects for games. The sound circuitry is capable of background operation, so that the BASIC can continue the action in games while a sound is being synthesised.

A major plus of the SV-318 is its expandability; a range of plug-ins and accessories is available that would make many other manufacturers green with envy. These range from a wide selection of games to an expansion chassis which will accept an RS-232 port, Centronics interface, extra RAM, a disk controller and 80-column card. Other options include a graphics tablet, games keypads and joysticks, data cassette recorder and dot matrix printer.

When used with the optional disk drives, the SV-318 uses the CP/M operating system, so that a wide range of software is available. The disks have a capacity of just over 160 Kbytes (formatted).

Setting up the SV-318 is just a matter of plugging in the power supply and modulator and wiring it to the back of the TV set. Immediately you're up and running. Attaching more of the peripherals and options might lead to a rat's nest of wiring, but this isn't a problem providing you don't want to move it all too often.

The SV-318 has an extended version of Microsoft BASIC, which includes all the usual graphics commands and the music macro language similar to that found in the IBM PC. It also includes provision for interrupt handling.

The manual for the SV-318 is nicely produced, and teaches the user BASIC through the use of the graphics statements – certainly a more interesting method than the usual mathematical approach. It progresses nicely until the end, when more complex statements, such as the SPRITE$ statement, are demonstrated by example but not explained. The user is left to try to deduce how the statement works.

The SV-318 is a nicely put together system; it has enough expansion capability to satisfy a wide range of requirements, and would be a good choice for the person who knows he wants a personal computer, but can't decide what for . . .

Micro-Professor II

There has been much talk recently about so-called 'rotten Apples' and Apple look-alikes. We have generally kept quiet on the subject; the legal complexities of registered designs make it a tricky subject, and even where such designs are legal, we feel it is very much a case of *caveat emptor*. These machines may suit some buyers, but in general, would you buy a car which claimed to be a copy of the Holden Commodore yet sold for only $3000?

One machine which seems to be able to stand on its own merits, yet offers a high degree of Apple compatibility, is the Multitech Micro-Professor II. This little (175 by 240 by 30 mm) box contains a 6502 microprocessor with 64 Kbytes of RAM and 16 Kbytes of ROM containing a monitor program and BASIC interpreter which accepts Applesoft BASIC programs.

The screen display looks just like the Apple's and the memory map (where the ROM fits, and where the graphics memory is) also looks just like that of the Apple. The Micro-Professor II can read Apple cassettes and, with the addition of a disk drive, floppy disks.

On the top front of the box is a 49-key calculator-style QWERTY keyboard, which – as with most of these keyboards – will horrify a touch typist. The tiny keys are closely packed with a non-standard layout.

The left-hand side of the case has a connector for a games cartridge, a Centronics printer port and a remote keyboard or joystick port. At the rear is a 50-pin connector which looks remarkably like an Apple expansion slot, but is subtly different and could not be relied on for full compatibility.

The major use of this slot is for the Micro-Professor II disk controller card, which connects the disk drive to the computer. The disk system is supplied with an MDOS II disk operating system, which bears more than a passing resemblance to Apple DOS 3.3 – but is obviously different in several ways.

The Micro-Professor II display is composed of 25 lines of 40 characters each, upper case only, just like the original Apple II and II+, and it provides lo-res and hi-res graphics in just the same manner. However, it is not completely Apple-compatible, and some mention of the differences may be in order.

While the Micro-Professor II can accept Applesoft programs, many such programs perform PEEKs and POKEs of memory locations associated with the monitor and graphics routines. These locations are different on the Micro-Professor II, and so such programs will not work, as a rule. However, if you understand the purpose of these PEEKs and POKEs, you can probably rewrite the program to work on the Micro-Professor II.

When it comes to machine code programs – the vast majority of good fast action games – the situation is even worse. These programs always use monitor routines and I/O port addresses to perform their I/O, and there's no way you can find those references and change them to work on the Micro-Professor II. The original author, who has the source code, could do it – if he thought it was worthwhile.

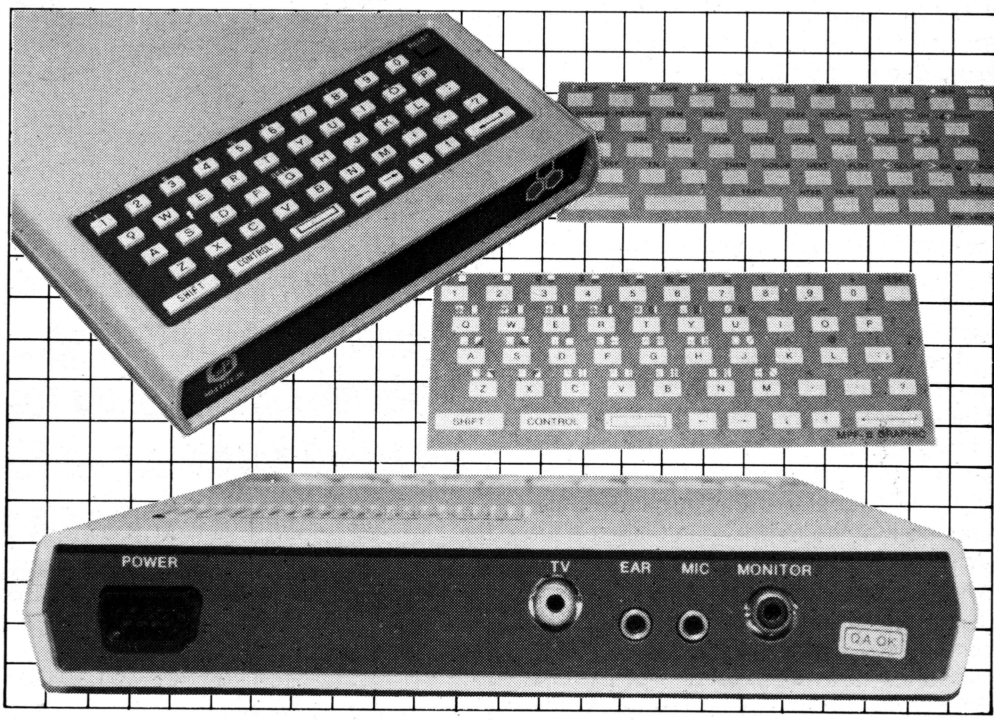
In summary, the Apple compatibility of the Micro-Professor II is probably of most use to someone who has already acquired a lot of experience with the Apple and is fully conversant with its operation. Such an individual could probably rewrite his/her own software to run on the MP without much difficulty. There's certainly not enough information in the Micro-Professor II manuals to get along without some of the Apple documentation as well, particularly when trying to use the graphics features such as shape tables.

In short, less than 100 percent compatible is not compatible; being slightly incompatible is like being slightly pregnant. And of course, where Apple is concerned, being 100 percent compatible is dangerous!

Someone should tell Multitech that quantity of documentation is no substitute for quality. The User's Manual and Introduction to BASIC Programming are reasonably well organised and take a good stab at providing plenty of reference material and technical background. The trouble is the translation into decidedly non-idiomatic English.

Thus we are presented with interpretation problems: what does the translator mean? What is a 'straight-thinging person'? How about 'Even if I were presented with a computer as a gift. I would be troubled as to whether I had enough room in my study to for it'.

The Micro-Professor II is, nonetheless, an interesting little machine which offers the ability to take advantage of the huge amount of software published in magazines and books. It also offers good value for money. It can run some quite good games cartridges and be a lot of fun – but I certainly wouldn't try to replace an Apple as a small business machine with a an MPF-II.



COMX 35

The COMX 35 is a very interesting machine indeed. For one thing, it's the first personal computer (other than simple single-board types) I've come across that uses the 1802A microprocessor. For another, it doesn't use Microsoft BASIC, but one that I've never seen before.

The COMX 35 has 16 Kbytes of ROM containing BASIC, and 32 Kbytes of user RAM, plus the screen RAM of 3 Kbytes. The display is 24 lines of 40 characters each, and eight different colours are available. The COMX display is unusual, in that computer output and echoed user input are displayed in different colours. As the user's manual points out, this is a useful feature for beginners.

The COMX 35 keyboard is a rubber type with a spongy feel, and a slightly non-standard (but still basically QWERTY) layout. At the right side of the keyboard is a built-in joystick. The computer has a built-in sound synthesiser and speaker.

Most micros use Microsoft BASIC; it's almost a novelty to find one that doesn't. COMX 35 BASIC is rather unusual. It is based on the ANSI standard, but with several extensions. Interestingly, it's an incremental compiler design. This means that when the user types RUN+, the interpreter does a scan through the program source code, and replaces all jumps to line numbers with jumps to an absolute address in memory.

This means that the program will run significantly faster, as much of a conventional interpreter's time is spent searching for the next line to be executed. Of course, if a program is edited, all the absolute addresses are changed, and so it will return to normal operation next time the program is run.

Graphics control is available in the COMX 35 through a user-definable character set, using special BASIC statements to manipulate it. This allows complete control over shapes and colours, including the creation of multi-colour shapes. The accompanying blurb also stated that the COMX 35 had "enhanced graphics developed along the Logo language" which I presume means turtle graphics, but I could find no mention of this in the manual and wasn't able to try it.

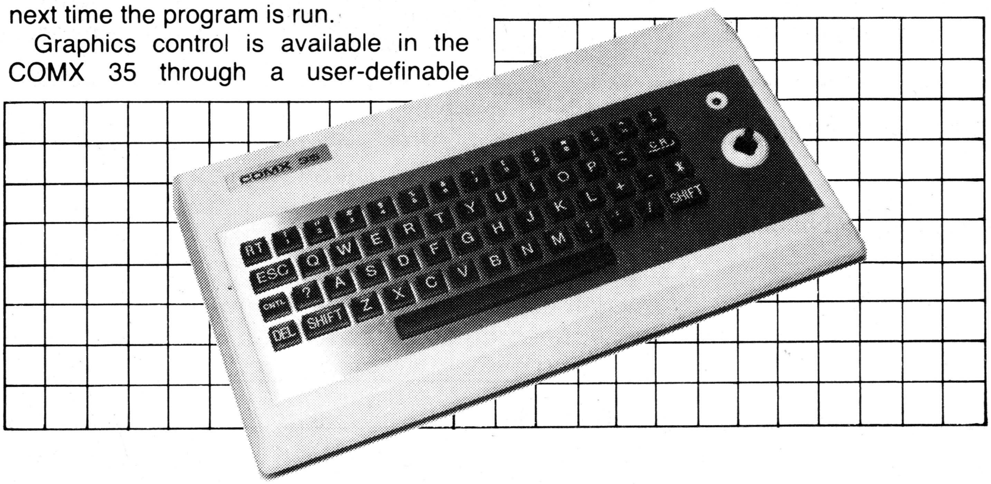
Other features of the language include the ability to set timer interrupts – great for game design – and the ability to save the entire data area of a program onto cassette. This presumably does away with the need for data files, although it means that data sets are restricted to the size of memory.

For those who find BASIC a bit limiting, the COMX 35 will also run Pascal or FORTH, so collectors of linguistic esoterica will be happy.

Various options are available for the COMX 35, including plug-in ROM packs, RS-232C and parallel printer interfaces, and a disk controller and drives. A speech synthesiser is also available.

Many programs are available for the COMX 35, including an electronic spreadsheet, simple databases, financial and statistical functions, a range of education programs and, of course, games – heaps of them – such as Othello, Hangman and various shoot'em-up, eat'em or catch'em variants.

An interesting machine, this; perhaps it will appeal best to the buyer who is happy to write his/her own software and will never want to key in programs straight from magazine pages without conversion. It's certainly an interesting one. . .



PERSONALLY... IT HAS TO BE NEC

NEC's desktop Personal Computers are a sophisticated and practical solution for any business that needs a personal support system.

They're designed for absolute reliability and built by NEC to the highest technical standards.

The APC (Advanced Personal Computer) is our revolutionary new desktop, just released in Australia.

Its integrated storage capacity (2MBytes) makes it the micro-computer that's best suited to a real desktop business environment.

The true 16-bit capability of its 8086 processor makes it the most powerful machine in its class.

The APC's high-resolution colour graphics (640 x 475 pixels) run circles, arcs and lines around everybody else, and its screen images set new standards in clarity.

Our business software has been optimised to match the APC's unique hardware features and capacity.

The wide range of software applications include business planning, budget analysis and all types of general accounting. There's also a remote intelligent terminal operation with asynchronous or bi-synchronous (3270, 3780) communications capability. All this from under $5,000 recommended retail price, including tax.

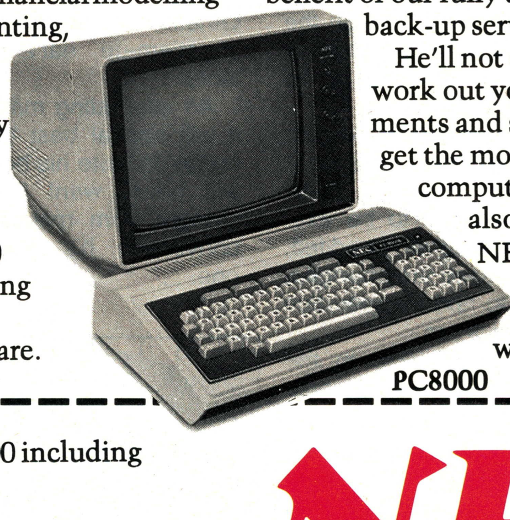
Both CP/M-86 and MS-DOS operating systems are supported by the APC, to solve business problems in the simplest, most cost-effective way possible.

The PC8000 desktop personal computer is our 8-bit little brother to the APC. It's ideal for financial modelling and general accounting, and has the same intelligent remote terminal capability as its big brother.

The PC8000 supports industry standard CP/M-80 and N-BASIC, giving it a wide range of applications software.

When you install an APC or PC8000 computer you will also get the benefit of our fully trained dealer back-up service.

He'll not only help you work out your exact requirements and show you how to get the most out of your computer system, he'll also show you how NEC stands behind him in the provision of hardware maintenance.



PC8000

Please send further information about the ☐ APC ☐ PC8000 including a list of authorised NEC dealers.

NEC Information Systems Australia Pty Ltd, 99 Nicholson Street, St. Leonards 2065. Telephone: 438 3544.

NEC

NEC CORPORATION
TOKYO JAPAN

Name: ____________________

Address: ____________________

____________________ Postcode: ____________________

NEC 526.

Commodore VIC-20

Well, what can one say? This is the machine that really turned Commodore around; it was getting a bit staid with the old PET series of machines, but the VIC really breathed life into the company.

The VIC has 5 Kbytes of memory – not a lot, these days – and can display up to eight colours. The memory is expandable to 32 Kbytes, using an external expander, and there is a plug-in ROM socket at the back.

Perhaps the best feature of the VIC is its full-sized keyboard, which even a fussy typist like me finds enjoyable to use. The keys are labelled clearly with alternative meanings like the graphics shapes and colours. At the side is a games port for a joystick, while the rear of the case boasts a row of cassette, user, serial and video ports.

The range of plug-in cartridges for the VIC is tremendous, and they really show the high-resolution graphics capabilities of the machine off to good advantage. The user can get at them with a super expander cartridge, which gives a 176 by 176 resolution.

The VIC's sound effects set a new standard in their time, with three voices of music. The whole package was quite revolutionary, and has done well in the intervening years – just look through the ads in this magazine . . .

Tandy Color Computer

This beastie reached our shores in late 1981, and provided a look at an alternative way of putting together a home computer. The Color Computer boasts a resolution of 256 by 192 under Color Extended BASIC, with nine colours, and as with the VIC, some of the games really make good use of that display.

The standard amount of ROM is 16K (with Extended BASIC) and most machines will have either 16 Kbytes or 32 Kbytes of RAM. The microprocessor used is the Motorola 6809E, an exceptionally powerful chip that is used to do a lot of the legwork inside the CoCo and save on hardware costs.

The CoCo can be expanded with a disk system, and many TRS-80 Model I programs could be run on it with virtually no modification, so it has quite a lot going for it.

The CoCo never really caught on in a big way (probably because the VIC was just so much cheaper), which is a shame as it really is a nice machine. It has a dedicated user group which does all kinds of weird and wonderful things with it and it really has a high level of support behind it, both from Tandy and elsewhere.

Texas Instruments 99/4A

The 99/4A is rather a problematical machine. It's almost sent Texas Instruments broke – figuratively speaking – yet it is a nice machine with all the things we are told a home computer ought to have.

The 99/4A has a 16-bit processor (though its BASIC is unaccountably slow) and a special graphics processor chip which looks after the 256 by 192 graphics and drives the sprites (moveable shapes) around the screen. The 'A' model features a decent keyboard compared to the earlier model.

A user-definable character set allows the user to create chess pieces, card symbols and other shapes.

The TI machine's manuals are well written, and TI arranged for additional material to be published by traditional publishing companies. Perhaps the greatest effort has been poured into software, with the release of a vast range of games, home applications and utilities, including TI Logo, a full implementation of that most interesting language, which really takes advantage of the graphics capabilities.

Other options include a speech synthesiser, expansion box, memory, disk drives and various I/O boards.

The machine has a definite following, with a large national user's group which has branches all over Australia. It's well supported and has great appeal.

Sinclair Spectrum

Worthy successor to Uncle Clive's ZX-80 and ZX-81, the ZX Spectrum is one of the tiniest personal computers around. It's hard to obtain in Australia, as the United States market apparently has first claim on production, but hopefully the situation will ease with time.

The Spectrum has a rubber-type keyboard, which has so many functions, symbols and letters it can be rather confusing at first. And, of course, it has a colour display, a major leap forward over the old '81. Eight colours are available, in two intensity levels.

For those who keep filling memory with their programs, good news: the Spectrum has 16 Kbytes of RAM as standard, and that is factory upgradeable to 48 Kbytes. The BASIC interpreter is in a 16 Kbyte ROM.

The Spectrum BASIC is a superset of the ZX-81 BASIC, with some additional statements. These include statements to change the colour of the border, the background and the foreground, as well as invert the colours, flash, and draw lines, circles and arcs.

The user can define his own character set by using the BIN statement, which allows him/her to specify which points to turn on in an eight by eight matrix.

The biggest let-down about the Spectrum is its sound capability – or lack of. It can synthesise a single tone through software, which of course stops the action while it makes sound effects.

The major form of mass storage is a cassette, but low-cost 'micro-floppy' drives are available – in fact these use a stringy-floppy style of tape wafer. An electrostatic printer is also available, which can print graphics off the screen.

The Spectrum has achieved remarkable success in Britain, where it seems almost everybody has one. Because of this, there is a fantastic amount of software appearing for it, plus books, magazines and the general support a home computer owner needs. It's going to be a successful machine.

BETTER WRITING–FAST!

How to get the word on the tip of your tongue into the text – without losing your flow.

Good writing is not always easy. And finding just the right word is often the hardest part. You could use a thesaurus – but who wants to leave the keyboard and wrestle with a 500 page book? So you go with what comes to mind. And your writing suffers.

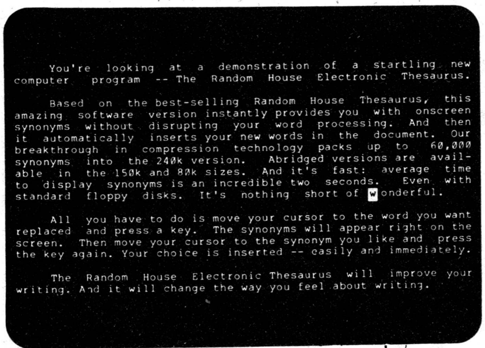
No more! Thanks to the Random House Electronic Thesaurus. As the screen illustrations show, you can now have instant, on-screen access to more than 60,000 synonyms indexed to the 4,700 most used (and overused) words in English. Without exiting your word processing program. Without waiting more than two seconds, on average. Without giant hard disks. Without complex directions. And your selections are automatically inserted into the text.

The Random House Electronic Thesaurus fits all systems. The 60,000 synonym version requires 240K and an abridged version is available for 150K (upgradable to the larger size). The software is totally compatible with WordStar 2.26, 3.0 and later. Or you can use it directly from CP/M.

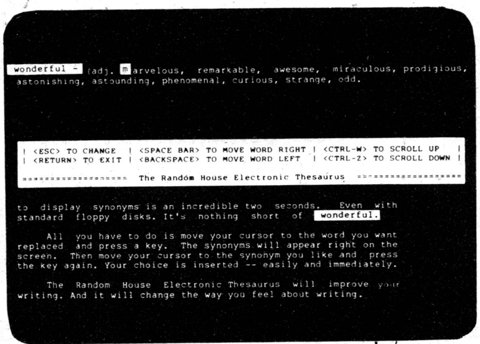
The Electronic Thesaurus is based on the popular Random House Thesaurus, edited and backed by one of the most respected names in publishing.

It will change the way you feel about writing. The price is right too, at just $250.00 – post and packing included.

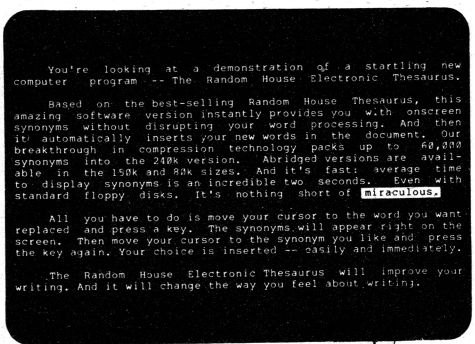
Three simple steps to increased word power:

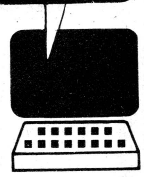


1 To find a synonym, simply position your cursor on the desired word. Or you can just type the word.



2 When you press ESCAPE twice. the top of the screen clears and fills with up to 100 synonyms. When you find a suitable synonym, bring your cursor to it and press ESCAPE again.





3 Your old word is deleted and the synonym is automatically inserted. Or you can press RETURN and return to your text without making any changes. You can also scroll through the listings.

Fill out and send in the coupon or simply ring us on (07) 221 9640 with your credit card number and we'll do the rest.

AUSTRALIAN DISTRIBUTORS:

THE GENESIS SOFTWAREHOUSE

8th Floor, City Mutual Building, 307 Queen Street, Brisbane 4000

To: THE GENESIS SOFTWAREHOUSE
8th Floor, City Mutual Building, 307 Queen Street, Brisbane 4000.

Please send me the Random House Electronic Thesaurus.

Name ______________________

Address ______________________

Cheque for $250 enclosed – or debit my credit card:

American Express)
Bankcard) No. ______________________
Diners Club) (Please delete the cards that don't apply)

Card expiry date __________ Signature ______________________

Not being a great game player, Les sought out an expert to deliver judgement on Dick Smith's Wizzard games machine. He dragged Mark Burnicle out of the pub for long enough to write this report . . .

SOME TIME ago, at a somewhat hard to remember party, Les Bell turned to me and said something about me writing an article for *Your Computer*.

Upon my enquiring on what the heck I would be able to scribble about, Les muttered 'Wizzard'. This brought an instant quizzical look from me as I'd been called many things, but wizzard – never!

'No, you can review the Wizzard!' he beamed at me, presenting me (almost magically) with a black cardboard box about the size of a beer carton and splashed with colour. Although not having a clue what I was holding, but not wishing to appear ignorant, I returned his excited banter about the black box with the pictures of Dick Smith, and muttered the word 'Wizzard' several times.

Eventually all appeared fine so, black box tucked safely under my arm, I set off to (as it turned out) review the Dick Smith Wizzard.

Once out of its package the Wizzard takes on the appearance of a quite simple, compact unit, complete with two joysticks, touch pads and firing buttons.

I decided, in a fit of irrational confidence, to connect the cords to their various points; but when all my efforts had failed, I reverted to the instructions and was soon under way.

To become accustomed to the machine I opted for a familiar invaders-style game called Sonic Invader. Since I was quite proficient at this style of game and had kept one particular hotel in Ultimo from going out of business with my patronage of its machine I decided this would provide a good idea of the Wizzard's ability.

To warm up I decided on battle number one of 16, a singles game which is Invaders at its most basic. The invaders move slowly from side to side, dropping slow moving tracers at you as they get cut to shreds. Definitely a good warm up. So on to battle number five, another singles battle – this time, however, the little devils kept disappearing. This was a bit tougher but I still managed to rack up points without much trouble.

At this point, full of confidence, I tried number 13, the hardest singles battle. The invaders move very quickly, rain you with bombs and disappear. This took a few goes before I managed to get the idea, but I soon had them under control.

I was impressed. The graphics were as good as the pub and milk bar versions, the controls moved quickly and accurately and it performed as well as you could hope for.

The doubles games were extremely

challenging, with your opponent firing at exactly the same time as yourself

Having warmed to the controls and the enjoyment of the competition, I tried the next game, Planet Defender, a variation on another of the standard games found in pubs and the like.

It Gets Harder . . .

Planet Defender finds you appearing from hyperspace to confront aliens which bear a remarkable similarity to bats and Halloween pumpkins. The game itself is a step up in ability and reflexes. Not only do you get to eradicate these little nasties on sight, you also plan your defence with a 'radar' device at the top of the screen. This is a game which tests the reflexes to the limit at the maximum difficulty stage.

The multi-colour bats and the green Halloween pumpkins glide in and out and then engulf your craft until you disappear in a thousand little pieces. A good game with very good graphics, the normal invaders type sound effects – though on the Wizzard these don't sound tinny but clear and crisp.

Finally (around 2 am the first time I played) I tried Tennis. This one really got me in. No longer is a video game of tennis restricted to two hyphens moving back and forth preventing a full stop from going past – no way! Things have progressed slightly.

I switched on the set and fell off my chair (*drunk again, Burnicle* – **Ed**) – there before me was a tennis court complete with net, crowd, Wimbledon sign (why settle for less?) and two little tennis players complete with racquets who run around actually hitting the ball over the net. The graphics are so damn great that the ball even casts a shadow.

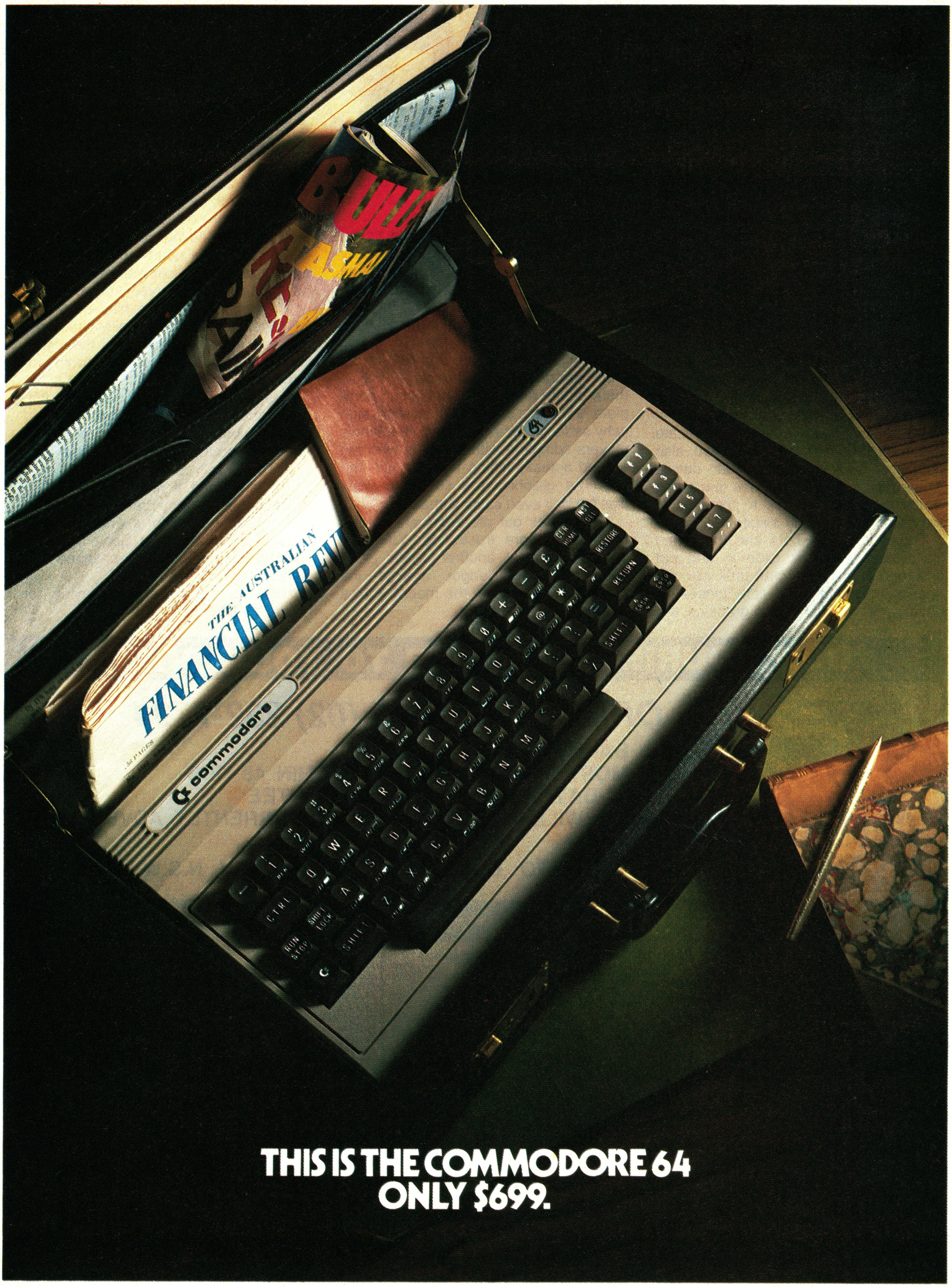
The games got progressively harder, faster and more frustrating. You may compete against another player or (if you like a challenge) against the machine. Believe me, the machine likes to win.

Playing an early level game I had it under control: serve, volley, volley again and the machine hits the ball right into the net. So, feeling capable of wiping McEnroe off the court I progressed up a few grades and followed my successful game plan. Serve, volley, volley again, and the machine hits a top spin lob over my player. While the early games play fairly predictably, the higher level games appear to hit at random patterns.

This game will turn you on (for hours), it will enthrall your little sister, your big brother, your mum, your dad and anyone else you may happen to show Wizzard Tennis to.

But be warned, if you sit down and try to beat the machine before you go to bed, you'll be there for a while. You will need a good supply of Scotch, a very comfortable chair and the patience of a saint.

This one will have you playing for hours. It's a Wizzard. □


THE AUSTRALIAN
FINANCIAL
commodore
THIS IS THE COMMODORE 64
ONLY $699.

BMS/CC22L

"THE COMMODORE 64 COULD BE THE MICROCOMPUTER INDUSTRY'S OUTSTANDING NEW PRODUCT INTRODUCTION SINCE THE BIRTH OF THIS INDUSTRY."

–SHEARSON/AMERICAN EXPRESS

The Commodore 64,™ to judge from the above comment, is generating as much excitement among the people who invest in companies as it is among the people who run them.

The reason for this is that, for the first time, high-level computer power is limited only by the inclination to have it, rather than the means.

HALF THE COST – TWICE THE PRODUCTIVITY

This simple equation reflects the Commodore 64's most basic – and outstanding – qualifications.

Its standard memory is 64K. Which is unusual enough in a micro at any price.

At $699, it is astonishing.

Compared, for example, with the Apple II+,® the Commodore 64 offers 33% more power at considerably less than 50% of the price.

Compared with anything less, it's even more impressive.

And it can effectively double your computer-equipped work force.

PILE ON THE PERIPHERALS

Because the basic cost of the Commodore 64 is so low, you can afford more peripherals for it. Like disk drives, printers or even printer-plotters.

For about the price of an Apple II+ computer alone, you could equip your company with a Commodore 64, disk drive and printer.

HARD FACTS ABOUT SOFTWARE

The Commodore 64 will run virtually any program important to business and industry.

Commodore software will include an electronic spreadsheet; business graphics (including printout); a user-definable diary/calendar; word processing; mailing lists, and more.

Even programs that can teach operators how to program.

It is programmable in BASIC.

With the added CP/M® option, your company will have access to hundreds of existing software packages.

In short, its applications are virtually limitless.

THE FUN SIDE OF POWER

The Commodore 64 can become very playful at a moment's notice.

You can use it with Commodore's game cartridges or invent your own diversions.

All will be enhanced by brilliant video quality and high resolution graphics (320 × 200 pixels, 16 available colors, 3D Sprite graphics), plus a music synthesizer.

NOW'S YOUR CHANCE

If you've been waiting for the "computer revolution," consider it as having arrived.

Through its 25 years of existence, Commodore has been committed to delivering better products at lower prices.

Today, the company's vertical integration has resulted in the Commodore 64's price performance breakthrough heralded by Shearson/American Express.

So if you run a company, you couldn't make a more cost-efficient capital investment.

For more information, contact your local Commodore dealer, or send in this coupon.

CP/M® is a registered trademark of Digital Research, Inc.

Commodore Business Machines Pty. Ltd.
5 Orion Road, Lane Cove NSW. 2066. (02) 427 4888.

Please send me more information on the Commodore 64.™

Name ______________________

Address ______________________

____

Postcode ____________ Phone ____________

BS

COMMODORE BUSINESS MACHINES PTY. LTD.
5 Orion Road, Lane Cove NSW. 2066. (02) 427 4888.

commodore
COMPUTER

BMS/CC22R

Pointing The Way To Pascal

In his on-going search for the perfect Pascal, Les Bell reviews a new native code compiler for the Z-80 processor under CP/M, COMPAS Pascal by Poly-Data.

PASCAL HAS achieved wide acceptance within just a few years in both the micro and minicomputer communities. Its only serious rival for the creation of complex yet maintainable code is C, mainly due to that language's strong ties with the Unix operating system.

Pascal is a good first language for newcomers to learn, and I was therefore interested to take a look at a new version which combines a Pascal compiler with a program editor. This combination provides fast switching between compiler aborts and program fixing, and is ideal for beginners.

COMPAS Pascal is produced in Denmark by Poly-Data Microcenter ApS, which is represented locally by K J Computer Services, of PO Box 66, Mentone VIC 3194. COMPAS comprises a floppy disk of software plus a large binder of documentation.

Upon receipt of the package, the first job is to produce a working diskette from the master, and then store the master away in a safe place. After SYSGENning a copy of the CP/M operating system onto the working disk, it is ready for use.

Because COMPAS includes an editor program, it has to be customised for the user's terminal before it will work correctly. This means running the CONFIG program and either choosing the appropriate terminal from a menu, or, if it does not appear, supplying the program with the appropriate control sequences for the terminal.

Another option allows the user to change the editing function keys to suit his requirements. Although COMPAS is supplied with sensible defaults – for example, B for Beginning, N for Next screen – the owner of a Televideo 950, for example, might like to change the 'down' code to control-V, which is what the down arrow key on the 950 transmits. This allows the user to take full advantage of his terminal's features.

COMPAS In Use

Just typing COMPAS at the CP/M command line will fire up the program. I suspect that COMPAS traces its heritage back to the Chung-Yuen Tiny Pascal, as its organisation is memory-based. In other words, once the compiler is loaded, then both the program source code and object code are also resident in memory.

The traditional compiler is organised to read lines of source code from a disk or tape file, perhaps performing multiple passes over the text. The COMPAS way of doing things is very much faster and more convenient. There are no delays for disk accesses, so the compilation is very fast, and when error messages are generated, both the editor and the source code are already in memory, making it easy and quick to fix errors.

At the prompt line, the user can perform a number of tasks. He can load or save a source code file, edit the contents of memory, compile and run programs, locate errors in the source code, list the disk directory, log in new disks and other tasks.

The editor is fairly primitive as screen-

your computer

SOFTWARE REVIEW

oriented editors go, but it does the job with no major problems. It has two modes: move mode, which is used for cursor movements and commands such as find and replace, and edit mode, which is used for text insertion. My only gripe is that quick insertion – such as placement of comment braces around redundant write statements – are a bit awkward using edit mode, a) because of the mode swapping and b) because in edit mode, everything after the cursor disappears from the screen and only returns in move mode.

However, there are a number of features which make this editor particularly attractive to the Pascal programmer. First of all, it supports auto-indentation, something that is particularly nice for showing the block structure of Pascal programs. And secondly, the editor is closely linked to the compiler, so that when an error is found, the compiler will display an error message, then automatically enter the editor with the cursor positioned on the spot where the error was found.

Because compilation is so fast, this makes fixing bugs a virtual joy in comparison with some systems I have used.

Once a program has been debugged and compiled, it is run by simply typing run. In fact, even if the program has not been compiled, the system will automatically invoke the compiler and then run the resultant code.

COMPAS Pascal compiled programs are also reasonably fast. On the Eratosthenes' Sieve benchmark (*Byte*, January 1983), the COMPAS Pascal version, running on a 2.5 MHz Kaypro II, came in at 56.6 seconds, which is quite a respectable result. Moreover, that is with range checking enabled (the default); turning it off reduced the time to 37.4 seconds. This is in the same area as most of the C compilers available, although not as fast as Pascal/MT+ and some of the other, more expensive, compilers which perform extensive optimisation.

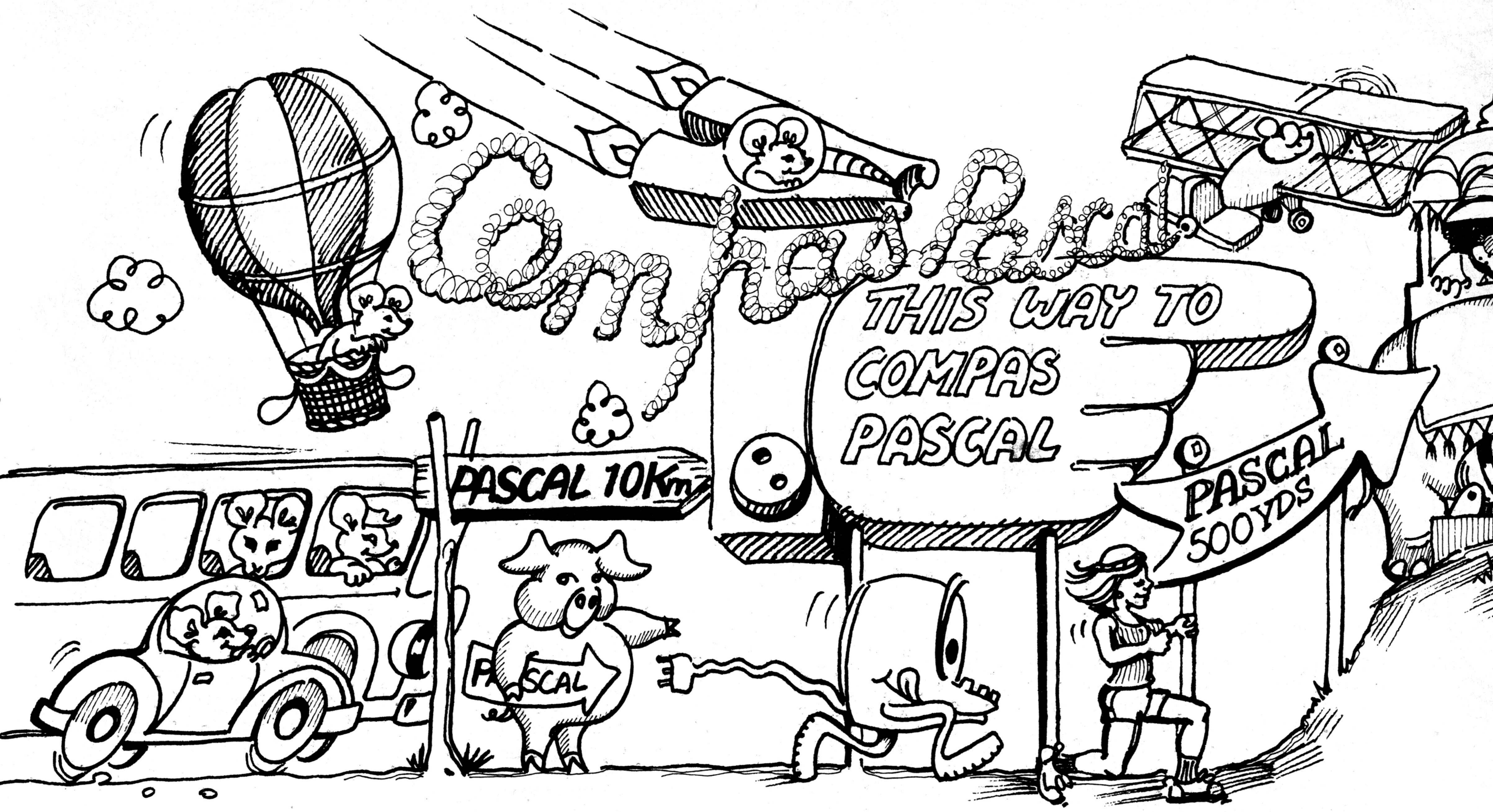
While the standard operating mode of COMPAS Pascal is to load source code under the COMPAS package, then compile it and run it using the same run-time module as the compiler itself, there are two alternatives. The PROGRAM command generates a CP/M .COM file containing both the run-time module and the generated object code, so that it can stand alone without the full COMPAS system. The OBJECT command generates an object code file without the run-time module; this file can only be loaded and run through the COMPAS Pascal chain procedure.

While the compiler will locate compile-time errors and pass their locations to the editor, run-time bugs cannot be fixed this way. COMPAS takes an interesting approach to the problem. The run-time module will produce an error message, giving the location of the error. The FIND command will then make a pass through the source code, keeping track of the addresses generated, until it finds the line which generated the offending code.

A similar command, WHERE, will perform the same function for included files.

The Language

COMPAS Pascal Revision 2.0 is a superset of Jensen and Wirth's Standard Pascal, with only a few minor differences from the standard (for example, mark and release rather than dispose, to maintain compatibility with UCSD Pascal).



The major difference many users will discover is the fact that the get and put procedures are not supported. Instead, the read and write procedures have been extended to handle all I/O needs.

The major extensions provided by COMPAS Pascal are dynamic strings, random access files, structured constants, logical operations on integers (naughty, naughty!), absolute address variables and in-line machine code generation.

Strings must be explicitly typed in the type declaration part of the program. For example:

```
type
  filename = STRING[15];
var
  infile : filename;
```

A selection of string functions and procedures is provided, including length, position of p within s, substring extraction and concatenation, deletion, insertion and others.

Random file I/O is possible through the seek procedure, which positions the file pointer to the specified record. The read and write procedures will operate on typed files as well as text files. COMPAS supports devices as files, so that references to CON:, for example, will read and write the console.

In addition, COMPAS Pascal can read and write blocks of data from untyped files. This provides a means of high-speed file transfer, as well as a solution to some tricky problems of reading 'foreign' files.

External procedures and functions are supported, so that machine code routines can be handled correctly. In addition, while the generated code normally uses the stack for storage of automatic variables (allowing recursion), it is possible to force procedures and functions to use absolute addressing of variables, providing tighter code which runs faster.

Since some Pascal programs may become too large to fit in the text editor buffer of COMPAS, the system provides the ability to read in 'include' files during compilation. This is particularly useful for creating libraries of useful subroutines which can be read in as required. What makes this facility particularly powerful is that, unlike standard Pascal, COMPAS allows free ordering and even multiple occurrences of the declaration sections of the program, so that library routines may declare their own types and variables.

An alternative way of splitting up programs so they will fit in memory is the use of chaining. This version of Pascal provides two types of chaining: firstly, chaining to COMPAS Pascal object files (.OBJ) without reloading the run-time package, and secondly, the execute procedure, which will load and run any CP/M command file.

Another useful facility is the ability to insert in-line machine code. While this does not include the capability of assembling the statements (they must be inserted as hex values), it at least avoids some of the problems some BASICs get into with string memory management.

One unusual facility provided by COMPAS Pascal is the ability to service interrupts, either in assembly language, or if required, in Pascal. An example is given in the manual of a real-time clock, based on a one-second interrupt to the Z-80.

The Good Books

The manual provided with the system is comprehensive, although the programmer will still need a copy of Jensen and Wirth. The manual is split into two parts: first, a user's guide to the editor and compiler, then the language specification.

The manual is terse, but quite readable, and peppered with good examples. It is also well organised for reference.

After briefly using COMPAS Pascal, my impressions of it are almost entirely good. It is a comprehensive implementation of standard Pascal, with additional functions and procedures to improve its utility in a microcomputer environment.

Despite being a native code compiler, it is not difficult to use, and it retains the highly interactive nature that impressed me about UCSD. At the same time, it possesses the raw speed of a native code compiler.

Previously, I had regarded Pascal/M as being a good choice for learning Pascal programming and Pascal/MT+ as being my choice for a production compiler. COMPAS Pascal will now displace Pascal/M as my choice for education, and I suspect that it has guts enough to function well in a production environment for all but the toughest jobs. □


```
PROGRAM HEXDUMP; {$A+,R-,S+}

{ This program will output a hex dump af any disk file. Each     }
{ line shows the address of the first byte of the line, then     }
{ 16 bytes in hex, and then the same bytes in ASCII, if they     }
{ are printable. On running the program, you must specify the    }
{ name of the disk file to be dumped, and the name of the disk   }
{ file or logical device (CON:, LST:, etc.), which is to reci-   }
{ eve the dump. The dump can be halted temporarily by pressing   }
{ any key, or aborted using ^C.                                  }

LABEL BREAK,EXIT;

TYPE
  SECTOR = ARRAY[Ø..7,Ø..15] OF BYTE;
  HEXSTRING = STRING[4];
  FILENAME = STRING[14];

VAR
  ADDRESS,L,P,I,J: INTEGER;
  CH: CHAR;
  INNAME,OUTNAME: FILENAME;
  BUFFER: SECTOR;
  INFILE: FILE;
  OUTFILE: TEXT;

FUNCTION HEX(NUMBER,DIGITS: INTEGER): HEXSTRING;
CONST
  HEXDIGIT: ARRAY[Ø..15] OF CHAR = 'Ø123456789ABCDEF';
VAR
  D: INTEGER;
  H: HEXSTRING;
BEGIN
  H[Ø]:=CHR(DIGITS);
  FOR D:=DIGITS DOWNTO 1 DO
  BEGIN
    H[D]:=HEXDIGIT[NUMBER AND 15];
    NUMBER:=NUMBER SHR 4;
  END;
  HEX:=H;
END;

BEGIN
  WRITE('Input file name? '); READLN(INNAME);
  IF INNAME='' THEN GOTO EXIT;
  WRITE('Output file name? '); READLN(OUTNAME);
  IF OUTNAME='' THEN GOTO EXIT;
  ASSIGN(INFILE,INNAME); {$I-} RESET(INFILE) {$I+};
  IF IORES>Ø THEN
  BEGIN
    WRITELN(INNAME,' does not exist'); GOTO EXIT;
  END;
  ASSIGN(OUTFILE,OUTNAME); REWRITE(OUTFILE);
  ADDRESS:=Ø;
  WHILE NOT EOF(INFILE) DO
  BEGIN
    BLOCKREAD(INFILE,BUFFER,1);
    FOR L:=Ø TO 7 DO
    BEGIN
      WRITE(OUTFILE,HEX(ADDRESS,4),' ');
      FOR P:=Ø TO 15 DO
      BEGIN
        IF P=8 THEN WRITE(OUTFILE,' ');
        WRITE(OUTFILE,HEX(BUFFER[L,P],2):3);
      END;
      WRITE(OUTFILE,'  ');
      FOR P:=Ø TO 15 DO
      BEGIN
        CH:=CHR(BUFFER[L,P]);
        IF (CH>=@32) AND (CH<=@126) THEN
        WRITE(OUTFILE,CH) ELSE WRITE(OUTFILE,'.');
      END;
      WRITELN(OUTFILE);
      ADDRESS:=ADDRESS+16;
      IF KEYPRESS THEN
      BEGIN
        READ(KBD,CH); IF CH=^C THEN GOTO BREAK;
        READ(KBD,CH); IF CH=^C THEN GOTO BREAK;
      END;
    END;
  END;
  BREAK:
  CLOSE(INFILE); CLOSE(OUTFILE);
  EXIT:
END.
```

```
PROGRAM PRIMES; {$R-}

{ This program will calculate and display all prime numbers }
{ between 1 and 3ØØØØ. The algorithm of the program is to    }
{ start out with an array containing representatives for     }
{ all odd numbers between 3 and 29999. Starting from 3 and   }
{ working upwards, each odd number is then tested. If a      }
{ number is still a member of the list when it is tested,    }
{ it is a prime number, and thus it is printed, and all odd  }
{ multiples of the number are eliminated from the list. As   }
{ can be expected, the program is quite slow on calculating  }
{ the very first primes, but from then on it gets faster     }
{ and faster. Note that 1 and 2 are assumed to be primes,    }
{ and not actually calculated.                               }

CONST
  MAX2 = 15ØØØ;    { MAXPRIME/2 }
  MAX3 = 1ØØØØ;    { MAXPRIME/3 }

VAR
  I,J,K: INTEGER;
  TEST: ARRAY[2..MAX2] OF BOOLEAN;

BEGIN
  WRITE(1:8,2:8);
  FOR I:=2 TO MAX2 DO TEST[I]:=TRUE;
  FOR I:=2 TO MAX2 DO
  IF TEST[I] THEN
  BEGIN
    J:=I+I-1; WRITE(J:8);
    IF J<MAX3 THEN
    BEGIN
      K:=I+J;
      WHILE K<=MAX2 DO
      BEGIN
        TEST[K]:=FALSE; K:=K+J;
      END;
    END;
  END;
  WRITELN;
END.
```

ZORBA

The Portable Computer...

...with 80 x 25 legible display and twin 400K disks.

Here are seven important reasons why ZORBA must be your best choice.

- 7" CRT.
- 80 x 25 Green Display.
- 400K Disks (single or double density).
- 19 programmable function keys.
- Disk emulations.
- Terminal emulations.
- Rugged construction.

Just released in Australia by InSystems, Zorba is set to be one of the top selling portables. Z80, CP/M based, Zorba has read/write and format compatibility with Xerox 820 (SD, DD) KAYCOMP (DD), DEC VT-180 (SD) and IBM PC (CP/M86).

The start up package for only $2995 (plus tax) includes CP/M 2.2 OS, M80 (inc. L80, LIB80, CREF80). Source code of BIOS and utilities, Data Comm. set-up package, Modem Comm. package, C Basic, WordStar, MailMerge, and CalcStar.

Total package backed by InSystems service and support and full 90-day warranty.

Dealer inquiries welcome.

insystems

337 Moray St., South Melbourne, 3205.
Telephone: (03) 690 2899.
Suite 11/84 Pacific H'way, St. Leonards, N.S.W. 2065.
Telephone: (02) 439 3788.

How to update Les Bell's year-old telephone directory program.

A Better Mousetrap...

By Lloyd R Borrett

AFTER TAKING delivery of my IBM-PC microcomputer, I had plenty to do. There were manuals to read and examples to practise with. Finally, I sat down and attempted to program the beast.

Unfortunately, the only programming language available was BASIC, and the IBM-PC dialect was new to me. After searching through magazines for a suitable example, I decided to implement an indexed telephone directory program, written by *Your Computer* editor Les Bell. The program had been developed for the "BASIC for Birdwatchers" tutorial in the July 1982 edition.

It proved to be a simple task to convert the program for the IBM-PC. However, I wasn't pleased with the program's structure and, once running, some operational limitations became obvious.

I proceeded to created an improved version of the program by changing to a more efficient sort algorithm, modifying the user interface, making allowance for duplicated names and re-structuring the program to use more sub-routines.

Code that is written in BASIC can be difficult to understand and maintain at the best of times. Effective use of the GOSUB statement is one of the easiest ways to improve the structure of a BASIC program.

The GOSUB statement can also reduce the amount of duplicated code. For example, the code to display a file record is duplicated in four places in the original program.

The major sub-routines I've used are:

620 ' *** Add a name to file ***
1090 ' *** Sorted list of file to screen ***
1170 ' *** Display specified names ***
1360 ' *** Delete name from file ***
1590 ' *** Find comment ***

The minor sub-routines used are:

1740 ' *** Routine to display a record ***
1850 ' *** Binary Search ***
2040 ' *** Sort KEYS$ and POSN ***
2180 ' *** Pack pointer arrays and file ***
2400 ' *** Find next record with same name ***
2510 ' *** Hold or Quit display ***
2580 ' *** Put uppercase 1$ in UPPER$ ***

The RETURN statement is only used as the last executable statement of these "sub-routines". This forces a 1-in, 1-out control structure.

In his original version of the program, Les Bell made no allowance for duplicated names. The binary search would always locate the same entry if there was more than one with the same name.

I have added some code which, upon finding a match, steps back through the index to find the first name that matches (lines 1940 to 1960). A pointer to that name is returned. It was then necessary to add a "sub-routine" to return a pointer to the next name that matches (lines 2400 to 2500). These "sub-routines" are used in the sections which find surnames (lines 1170 to 1350) and delete names (lines 1360 to 1580).

Les Bell described the sort used in the original version as a standard Shellsort. While it worked, it was s-l-o-w...

After consulting *Algorithms Plus Data Structures Equals Programs* (Niklaus Wirth, Prentice-Hall, 1976) and *The Art of Computer Programming, Volume 3, Sorting and Searching* (Donald Knuth, Addison-Wesley), I decided to try a shuttle interchange sort.

I set up an array of 250 randomly ordered names and ran the two procedures over them. The shuttle interchange sort took 493 seconds, and the Shellsort took 560 seconds. Only a minor improvement.

But in this program, the typical case is a few names being sorted into the existing list. So I ran a procedure which would add five names to a list of 250 sorted names.

The shuttle interchange sort took 20 seconds, and the Shellsort took 506 seconds. Now, that's some difference! I decided to use the shuttle interchange sort!

I can think of ways to speed up the sort even further, and it's possible that another algorithm may also prove faster, but for the time being I shall stick with what I've got.

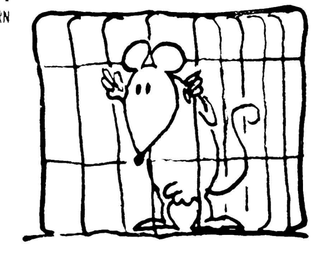
While there are many ways in which this program could be further improved, I've achieved the goals I set for myself when I started modifying it. So, for the time being, I'll concentrate on using it, instead of playing with it.

While working on a program, there comes a point in time when you feel any extra effort on your part can't be adequately returned. Les Bell probably felt he'd reached that point when he finished the original version.

I've reached that point now. How much further can *you* take it?

```
"teldir.bas", listed on 03-15-1983 at 13:26:13

 10 '          *** Address / Telephone Directory ***
 20 '   TELDIR Revision 1.00
 30 '   Written by Lloyd Borrett, March 1983.
 40 '   Based on a program by Les Bell, "Your Computer",
     July 1982.
 50 '   "BASIC For Birdwatchers", Part IX, page 54.
 60 '
 70 '          *** Initialize ***
 80 DEFINT A - Z
 90 DEF FNLN$(L) = STRING$(L,95)
     ' Used to produce a string of L underscores.
100 MAXREC = 256  ' Maximum number of records per file
110 TDELAY = 500  ' Time delay constant
120 NONE = 0
     ' No records constant, not in range 1 to maxrec
130 DELETED = NONE  ' Used to flag a record as deleted.
140 NO = 0         ' Constant value for NO
150 YES = 1        ' Constant value for YES
160 DIGO = ASC("0")
170 DIM KEYS$(256), POSN(256)

180 '          *** Open the files ***
190 '   Get the file name
200 CLS : LOCATE 3,1
210 INPUT "Enter the directory name: [TELDIR] ";FILE$
220 IF FILE$="" THEN FILE$="TELDIR"

230 '   Read the index file
240 ON ERROR GOTO 510
250 OPEN "I", #1, FILE$+".IND"
260 TOPREC = 1    ' Pointer to the next available record
270 IF EOF(1) THEN CLOSE 1 : GOTO 290
280 INPUT #1, KEYS$(TOPREC), POSN(TOPREC) :
      TOPREC = TOPREC + 1 : GOTO 270
290 LOCATE 5,1 :
      PRINT TOPREC-1;" entries in directory ";FILE$;"."
300 FOR I = 1 TO TDELAY : NEXT I

310 '   Open the directory file
320 OPEN "R", #2, FILE$+".TDR", 128
330 FIELD #2, 20 AS FSNAME$, 20 AS FCNAME$, 30 AS FADDR$,
     20 AS FCITY$, 4 AS FPSTCD$, 15 AS FTEL$, 19 AS
     FCOMNT$

340 '          *** Display Menu ***
350 SCREEN 0,1 : COLOR 7,0 : CLS
360 LOCATE 3,1 :
      PRINT "TELDIR Revision 1.00 on ";DATE$;" at ";TIME$
370 LOCATE 5,1 :
      PRINT "*** Address / Telephone Directory ***" :
      LOCATE 8,1
380 PRINT "    1. Find name          (Search directory for
     a surname)"
390 PRINT "    2. Find comment       (Search directory for
     a comment)"
400 PRINT "    3. View directory     (Display the full
     directory)"
410 PRINT "    4. Add name           (Create new directory
     entries)"
420 PRINT "    5. Delete name        (Remove directory
     entries)"
430 PRINT "    6. Exit               (Quit and return to
     DOS)"
440 PRINT : PRINT "Press choice: ";

450 '   Get and process the choice.
460 I$ = INKEY$ : IF I$ = "" THEN 460
470 IF I$ < "1" OR I$ > "6" THEN BEEP : GOTO 460
480 ON ASC(I$)-DIGO GOSUB 1180,1600,1100,640,1370,570
490 GOTO 350

500 '          *** Handle errors ***
510 BEEP : IF ERR <> 53 THEN 540
520 CLOSE :
      PRINT "Directory ";FILE$;
     " doesn't exist.  Creating it."
530 TOPREC = 1 : RESUME 320
540 LOCATE 23,20 :
      PRINT "*** Error";ERR;"in line";ERL;"." : STOP

550 '          *** Exit to DOS ***
560 '   Write index file
570 OPEN "O", #1, FILE$+".IND"
580 FOR I = 1 TO TOPREC-1 : WRITE #1, KEYS$(I), POSN(I) :
      NEXT I : CLOSE
590 REM
600 CLS : SYSTEM

620 '          *** Routine to add a name to file ***
630 REM
640 ADDFLG = NO

650 '   Is there room for more ?
```

```
660 IF TOPREC <= MAXREC THEN 710
670 CLS: LOCATE 5,1:
      PRINT "There's no room for more names in directory
      ";FILE
680 BEEP :
      PRINT "unless you delete some of the existing
      names."
690 FOR I = 1 TO TDELAY : NEXT I : GOTO 1070

700 '  Get another name
710 CLS : LOCATE 5,1 :
      PRINT "Create new directory entries." : LOCATE 7,1
720 PRINT "Surname     : ";FNLN$(20)
730 PRINT "First Name  : ";FNLN$(20)
740 PRINT "Street      : ";FNLN$(30)
750 PRINT "Town/City   : ";FNLN$(20)
760 PRINT "Postcode    : ";FNLN$(4)
770 PRINT "Telephone   : ";FNLN$(15)
780 PRINT "Comment     : ";FNLN$(19)

790 '  Get the details
800 LOCATE 7,13 : INPUT SNAME$ : IF SNAME$ = "" THEN 1070
      ELSE I$ = SNAME$ : GOSUB 2590
810 LOCATE 7,13 : PRINT ": ";SNAME$;SPACE$(20-LEN(SNAME$))
820 LOCATE 8,13 : INPUT CNAME$
830 LOCATE 8,13 : PRINT ": ";CNAME$;SPACE$(20-LEN(CNAME$))
840 LOCATE 9,13 : INPUT ADDR$
850 LOCATE 9,13 : PRINT ": ";ADDR$;SPACE$(30-LEN(ADDR$))
860 LOCATE 10,13 : INPUT CITY$
870 LOCATE 10,13 : PRINT ": ";CITY$;SPACE$(20-LEN(CITY$))
880 LOCATE 11,13 : INPUT PSTCD$
890 LOCATE 11,13 : PRINT ": ";PSTCD$;SPACE$(16)
900 LOCATE 12,13 : INPUT TEL$
910 LOCATE 12,13 : PRINT ": ";TEL$;SPACE$(15-LEN(TEL$))
920 LOCATE 13,13 : INPUT COMNT$
930 LOCATE 13,13 : PRINT ": ";COMNT$;SPACE$(19-LEN(COMNT$))

940 '  Write record to disk.
950 LSET FSNAME$ = SNAME$
960 LSET FCNAME$ = CNAME$
970 LSET FADDR$ = ADDR$
980 LSET FCITY$ = CITY$
990 LSET FPSTCD$ = PSTCD$
1000 LSET FTEL$ = TEL$
1010 LSET FCOMNT$ = COMNT$
1020 PUT #2, TOPREC

1030 '  Add record to index and go back to get more records
1040 KEYS$(TOPREC) = UPPER$ : POSN(TOPREC) = TOPREC :
      TOPREC = TOPREC + 1
1050 ADDFLG = YES : GOTO 660

1060 '  Finish off adding names.  Sort and return to the
      main menu.
1070 IF ADDFLG = YES THEN GOSUB 2050
1080 RETURN

1090 '          *** Sorted listing of file to the screen ***
1100 FOR POINTER = 1  TO TOPREC-1 :
      IF POSN(POINTER) = DELETED THEN 1150

1110 '  Get and display the record
1120  GET #2, POSN(POINTER) : GOSUB 1750
1130 '  Hold or quit
1140  GOSUB 2520 : IF QUIT = YES THEN 1160
1150 NEXT POINTER
1160 RETURN

1170 '          *** Display specified names ***
1180 CLS : LOCATE 3,1 :
      PRINT "Search the directory for a surname."
1190 LOCATE 5,1: INPUT "Enter surname to find: [Exit] ";I$ :
      IF I$="" THEN 1350

1200 '  Convert to all uppercase and search for it.
1210 GOSUB 2590 : SNAME$ = UPPER$ : GOSUB 1890 :
      IF POINTER = NONE THEN 1310

1220 '  Get and display the record
1230 GET #2, POSN(POINTER) : GOSUB 1750
1240 LOCATE 20,20 :
      PRINT "Press SPACE BAR to continue, ESC to quit.";
1250 I$ = INKEY$ : IF I$ = "" THEN 1250
1260 IF I$ = " " THEN 1290
1270 IF I$ = CHR$(27) THEN 1330 ELSE BEEP : GOTO 1250

1280 '  Any more with the same name?
1290 GOSUB 2440 : IF POINTER = NONE THEN 1330 ELSE 1230

1300 '  Unsuccessful search
1310 LOCATE 7,15: BEEP: PRINT SNAME$;" not found."; :
      FOR I=1 TO TDELAY : NEXT

1320 '  Go back for more names to display
1330 GOTO 1180

1340 '  Finished displaying names
1350 RETURN

1360 '          *** Routine to delete name from file ***
1370 DELFLG = NO
1380 CLS : LOCATE 3,1 : PRINT "Delete directory entries."
1390 LOCATE 5,1: INPUT "Enter surname to delete: [Exit] ";I$
1400 IF I$ = "" THEN 1570
1410 GOSUB 2590 : SNAME$ = UPPER$ : GOSUB 1890 :
      IF POINTER = NONE THEN 1530

1420 '  Display it
1430 GET #2, POSN(POINTER) : GOSUB 1750
1440 LOCATE 20,20 : PRINT  "Delete? (Y/N) "
1450 I$ = INKEY$ : IF I$ = "" THEN 1450
1460 IF I$ = "Y" OR I$ = "y" THEN 1490
1470 IF I$ = "N" OR I$ = "n" THEN 1510 ELSE BEEP : GOTO 1450

1480 '  Flag the record as deleted ?

1490 POSN(POINTER) = DELETED : DELFLG = YES

1500 '  Any more with the same name ?
1510 GOSUB 2440 : IF POINTER = NONE THEN 1550 ELSE 1430

1520 '  Unsuccessful search
1530 LOCATE 7,15: BEEP: PRINT SNAME$;" not found." :
      FOR I=1 TO TDELAY : NEXT I

1540 '  Go back for more names
1550 GOTO 1380

1560 '  Finish off deleting names.  Pack and return to the
      main menu.
1570 IF DELFLG = YES THEN GOSUB 2200
1580 RETURN

1590 '          *** Find comment ***
1600 CLS : LOCATE 3,1 :
      PRINT "Search the directory for a comment."
1610 LOCATE 5,1 : INPUT "Enter comment to search for:
      [Exit] ";I$
1620 IF I$ = "" THEN 1730 ELSE GOSUB 2590 : COMNT$ = UPPER$

1630 '  Test every directory record for the comment
1640 CLEN = LEN(COMNT$)
1650 FOR POINTER = 1 TO TOPREC-1 : GET #2, POINTER
1660   I$ = LEFT$(FCOMNT$,CLEN) : GOSUB 2590

1670 '  If it matches, display it
1680   IF UPPER$ = COMNT$
      THEN GOSUB 1750 : GOSUB 2520 : IF QUIT = YES
      THEN 1710
1690 NEXT POINTER

1700 '  Go back for more comments to search for
1710 GOTO 1600

1720 '  Finished searching for comments
1730 RETURN

1740 '          *** Routine to display a record ***
1750 CLS: LOCATE 5,1: PRINT "Entry:
      ";POINTER;"      Position: ";POSN(POINTER)
1760 LOCATE 7,1
1770 PRINT "Surname     :  ";FSNAME$
1780 PRINT "First Name  :  ";FCNAME$
1790 PRINT "Street      :  ";FADDR$
1800 PRINT "Town/City   :  ";FCITY$
1810 PRINT "Postcode    :  ";FPSTCD$
1820 PRINT "Telephone   :  ";FTEL$
1830 PRINT "Comment     :  ";FCOMNT$
1840 RETURN
1850 '          *** Binary Search ***
1860 '  Finds the first occurance of SNAME$ in
      KEYS$(MAXREC).
1870 '  The index position is returned in POINTER.
1880 '  Set upper and lower boundaries
1890 POINTER = NONE : LOWER = 1 : UPPER = TOPREC - 1

1900 '  Try the middle of the current interval
1910 IF UPPER < LOWER THEN 2030 ELSE I = (LOWER + UPPER) \ 2
1920 IF KEYS$(I) < SNAME$ THEN LOWER = I + 1 : GOTO 1910
1930 IF KEYS$(I) > SNAME$ THEN UPPER = I - 1 : GOTO 1910

1940 '  We have a match.  Get first of possible duplicates.
1950 IF I <= 1 THEN 1980
1960 IF KEYS$(I - 1) <> SNAME$ THEN 1980
      ELSE I = I - 1 : GOTO 1950

1970 '  Find first record not deleted.
1980 IF POSN(I) <> DELETED THEN 2020
1990 IF KEYS$(I) <> KEYS$(I + 1) THEN 2030
2000 I = I + 1 : IF I > TOPREC - 1 THEN 2030 ELSE 1980

2010 '  Put index position in POINTER and/or return
2020 POINTER = I
2030 RETURN

2040 '          *** Sort KEYS$(maxrec) and POSN(maxrec)
2050 CLS : LOCATE 5,1 :
      PRINT "Sorting the directory.  Please wait."
2060 J = 1
2070 IF J > TOPREC - 2 THEN 2160
2080 IF KEYS$(J) <= KEYS$(J+1) THEN 2150
2090 SWAP KEYS$(J), KEYS$(J+1) : SWAP POSN(J), POSN(J+1) :
      PRINT ".";
2100 I = J - 1
2110 IF I < 1 THEN 2150
2120 IF KEYS$(I) <= KEYS$(I+1) THEN 2140
2130 SWAP KEYS$(I), KEYS$(I+1) : SWAP POSN(I), POSN(I+1) :
      PRINT ".";
2140 I = I - 1 : GOTO 2110
2150 J = J + 1 : GOTO 2070
2160 PRINT
2170 RETURN

2180 '          *** Pack pointer arrays and file ***
2190 '  Pack pointer arrays
2200 CLS : LOCATE 5,1 :
      PRINT "Packing the directory.  Please wait." : D = 1
2210 FOR I = 1 TO TOPREC - 1 : IF POSN(I) = DELETED
      THEN 2240
2220   IF D <> I
      THEN KEYS$(D) = KEYS$(I) : POSN(D) = POSN(I)

2230   D = D + 1
2240 NEXT I

2250 '  Were any records deleted ?
2260 IF D = TOPREC THEN 2390

2270 '  Yes.  Pack file
2280 P = 1
2290 FOR J = 1 TO TOPREC - 1
2300   FOR I = 1 TO D-1 : IF POSN(I) = J THEN 2340
      ELSE NEXT I

2310 '  Record not needed
2320   GOTO 2360

2330 '  Record still needed
2340   IF P <> J THEN GET #2, J : PUT #2, P : POSN(I) = P
2350   P = P + 1
2360 NEXT J
2370 IF P <> D
      THEN PRINT "ERROR:
      File and pointer array do not match" : STOP
2380 TOPREC = D
2390 RETURN

2400 '          *** Find the next record with the same name
      ***
2410 '  Name is in SNAME$,  current position in index is
      in POINTER.
2420 '  Returns POINTER =
      NONE if no more records with the same name,
2430 REM otherwise POINTER points to a new position in the
      index.
2440 I = POINTER : POINTER = NONE
2450 I = I + 1 : IF I >= TOPREC THEN 2500
2460 IF SNAME$ <> KEYS$(I) THEN 2500
2470 IF POSN(I) = DELETED THEN 2450

2480 '  Found one
2490 POINTER = I
2500 RETURN

2510 '          *** Hold or quit display ***
2520 QUIT = YES : LOCATE 20,20 :
      PRINT "Press SPACE BAR to hold, ESC to quit."
2530 FOR I = 1 TO TDELAY : I$ = INKEY$ : IF I$ = ""
      THEN 2560
2540   IF I$ = " " THEN I = 1 : GOTO 2560
2550   IF I$ = CHR$(27) THEN 2570 ELSE BEEP
2560 NEXT I : QUIT = NO
2570 RETURN

2580 '          *** Put an all uppercase version of I$ in
      UPPER$ ***
2590 UPPER$ = I$ : IF LEN(I$) = 0 THEN 2640
2600 CONVERT = ASC("a") - ASC("A")
2610 FOR I = 1 TO LEN(I$) : II$ = MID$(I$,I,1)
2620   IF II$ >= "a" AND II$ <= "z"
      THEN MID$(UPPER$,I,1) = CHR$(ASC(II$) - CONVERT)
2630 NEXT I
2640 RETURN
```



THE PROFESSIONAL TERMINAL AT AN AFFORDABLE PRICE

The WY 100 intelligent terminal has so many features, at half the price you'd expect to pay. It's easy to justify your choice.

Here's why ...

- Fully tilting & rotating screen,
- Detached keyboard with 105 keys,
- Vertical & horizontal split screen capability,
- Data validation/protected forms,
- Character/block transmission,
- Programmable functional keys,
- 14" non-glare screen,
- Full 24 x 80 display, with extra line for message fields and function key labelling,
- Line drawing and business graphics,

* Now fully assembled, tested and supported in Australia. All for $1350 plus $190 tax if applicable.

Need any more reasons to choose Wyse? Ring us today.

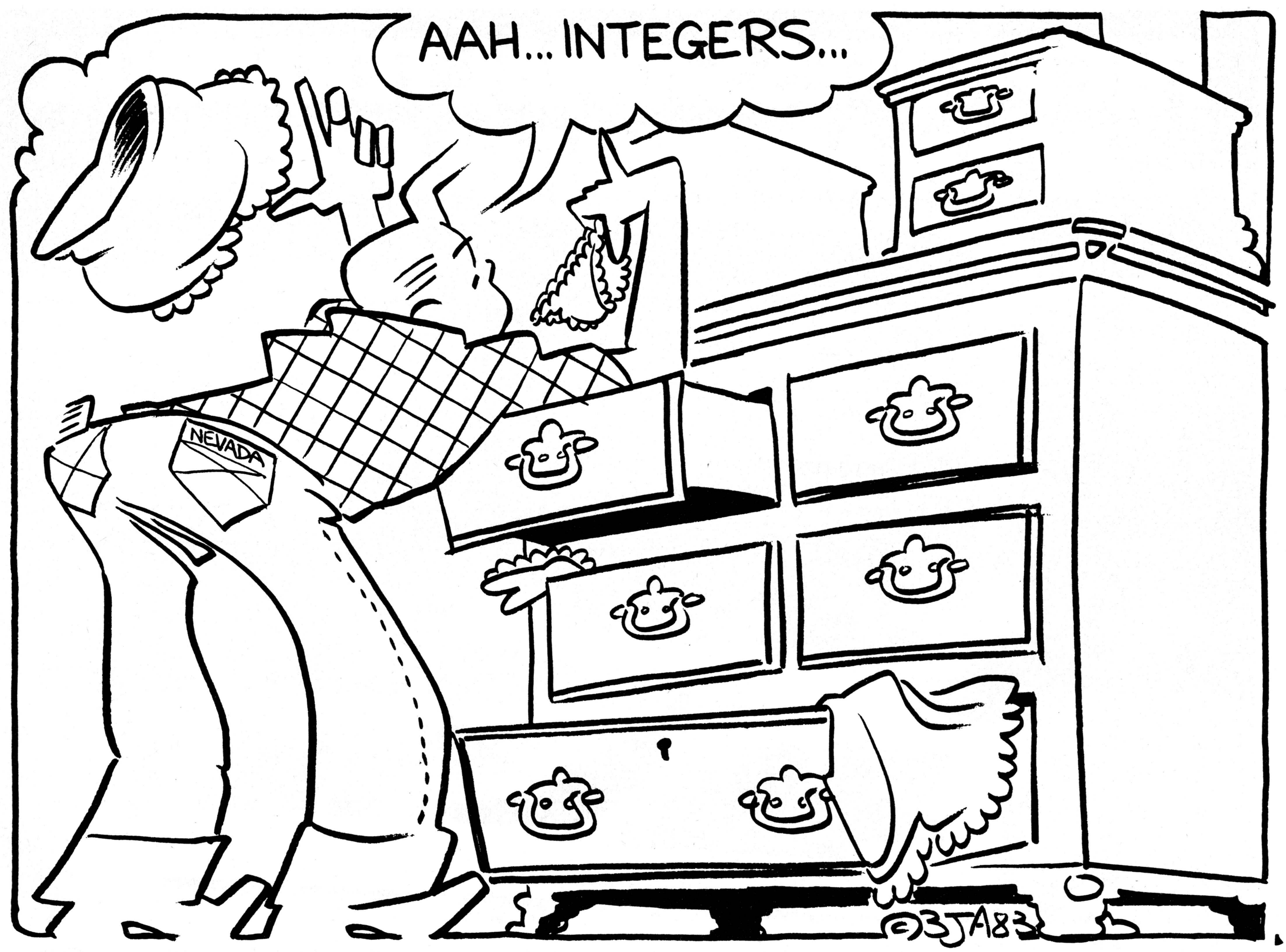
Preferred microcomputer supplier to Victorian Govt.

mpa Microprocessor Applications Pty. Ltd.

48 Rutland Rd., Box Hill, Vic. 3128
Tel: (03) 890 0277. Telex: AA 31187.

Dealers:
Sydney: Synax Systems (02) 439 7657.
Adelaide: Mentor Management (08) 42 4842.
Perth: Alpha Media Sales (09) 368 1459.
Brisbane: Paravet Instruments (075) 31 7656.

MPA543



Nevada Fortran And Mother's Drawers

Data-processing professionals, looking for a Fortran compiler, are apt to laugh when you suggest Nevada Fortran. However, at only $49.95, can they go (or be) wrong? Les Bell finds out...

THERE AREN'T many Fortran compilers around for microcomputers, and those that are available tend to be expensive beasts. Despite its late-Fifties genesis, Fortran is not a simple or small language, and this is reflected in the compilers.

Professional programmers often ask me where they can obtain a Fortran compiler to run under CP/M on an Osborne or similar computer. A lot of the time, the choice comes down to MicroSoft Fortran, UCSD p-system's Fortran-77, or Nevada Fortran.

When I suggest that, at just $49.95, they ought to check out Nevada Fortran first, buying a larger compiler if necessary, they sometimes suppress a snigger. You see, they're use to paying tens of thousands of dollars for compilers, and surely a fifty-buck cheapie can't be much good – can it?

Oh yes, it can. I'm no authority on Fortran – I was raised on Algol at school and have a natural aversion to nonstructured languages – but, good lord, you don't have to be a genius to recognise a bargain that is this good.

Nevada Fortran is an extended version of ANSI Standard Fortran, version X3.9-1966 (otherwise known as Fortran IV), with these exceptions: No double precision, no complex numbers, no equivalence statement, no initialisation in DATA statements, no statement functions, only the first five characters of names are retained, and the 8080 register names (A, B and so on) are reserved names for functions, subroutines or COMMON blocks, though okay for variables.

What is left is a very useable subset of full Fortran, certainly covering all that I used during my student days, and certainly well up to the requirements of the average engineer, scientist or other nonfulltime programming user. And for just $49.95!

Fortran was the first high-level language to catch on with more than one computer manufacturer. Designed by

IBM's John Backus during the late-Fifties, its name is a contraction of *For*mula *Tran*slator, and its major area of application is scientific and technical programming, though a few companies have been known to use it for commercial programming.

If you imagine BASIC with no need for line numbers, except as targets for GOTOs, horrendously complex formatted input/output statements instead of nice simple "input" and "print", format statements that look like Sanskrit in comparison with "print using", and an arithmetic "if" statement that causes more trouble than any other feature of any other language – that's Fortran.

How Do I Learn It?

People either love it or hate it. Those under 35 almost universally hate it – to paraphrase a recent truism, it's the kind of language only a mother could love.

Despite that, it does have some things going for it. For a start, it has made a lot of things possible for people that otherwise would have been impossible. And, as a consequence of that, there is a tremendous amount of scientific software written in Fortran which is virtually in the public domain, such as econometric models, graphics drivers for Tektronix terminals – you name it.

Fortran runs on just about every computer you can think of, from giant mainframes down through minis to micros. Unfortunately, most implementations are non-standard in some way or other, so it's not the most portable of languages, despite the existence of an ANSI standard.

I don't propose to turn this article into a tutorial on Fortran programming. Instead, I'm going to take the opportunity to review/plug an excellent book I came across some time ago. Called *A Fortran Coloring Book*, by Dr Roger Kaufman and published by the MIT Press, it is without a doubt one of the best programming texts ever devised. It never caught on with universities, primarily because its cartoon-style corny humour doesn't fit the establishment viewpoint of academia.

A Fortran Coloring Book proudly sports a money-back guarantee ("offer void within or without the United States"): "If you do all the programming problems at the end of this book and don't learn Fortran in the process, you deserve your money back." A couple more excerpts will serve to illustrate. The book begins: "A computer is like your Mommy's bureau drawers. It has big drawers for numbers with decimal points. These are called real or floating point numbers...and it has teeny tiny drawers for integer numbers. Integers are numbers without decimal points or fractional parts. Integers are used for counting and things like that." Or on the subject of punched cards: "There are 12 rows that go across each card. For reasons known only to Thomas Watson or God, the top row is known as the 12 edge and the bottom row is known as the nine edge. Thus, one sometimes sees instructions, such as the following alleged excerpt from Thomas Watson's will, which specifies that when he dies, he is to be buried 'nine edge face down'." (Thomas Watson founded IBM, by the way.)

Or how about this gem: "Algorithms are a little like logarithms in that neither one comes from the the Babylonian word, *Nosherai*, which Mesopotamian scholars have translated to mean 'Vestal Virgins'. However, that is where their similarity ends. Logarithms, as you well know, are a singing group at MIT. An algorithm, on the other hand (not *that* one – the *clean* hand!) is a procedure for solving a problem." Anyone who can survive the barrage of awful puns, corny jokes and crummy cartoons in this book will actually come out of it knowing a heck of a lot more about Fortran than when he started, particularly if the exercises at the end are attempted. I found it an excellent refresher course (particularly the flowchart for efficiently picking lint out of one's belly button).

For those who want to learn Fortran, Nevada Fortran plus *A Fortran Coloring Book* is an excellent course at somewhat under $70 all up.

For those who don't want to learn Fortran, but are curious anyway, here are a few Fortran programs: □

```
OPTIONS X
C
C GRAPH SINE FUNCTION FROM -PI TO PI IN INCREMENT OF .12
C
        DIMENSION LINE(70)
        INTEGER WHERE
C
C OPEN UNIT 6 TO WRITE TO CONSOLE
C
        CALL OPEN (6,'CON:')
C
C WRITE TITLE
C
        WRITE (6,2)
2       FORMAT (28X,'GRAPH OF SIN')
        TYPE
        TYPE
C
C SET PI AND -PI
C
        PI=3.1415926
        MPI=-PI
C
C MAIN LOOP
C
        DO 100 ANGLE=MPI,PI,.12
C
C FIGURE OUT WHICH ELEMENT IN ARRAY SHOULD BE SET TO *,
C SIN RETURNS -1 TO 1 WHICH IS CONVERTED TO -35 TO 35
C AND THEN OFFSET SO FINAL RANGE IS 1 TO 70
C
        WHERE=SIN(ANGLE)*35+35
C
C FIGURE OUT HOW MUCH TO BLANK IN THE OUTPUT ARRAY
C
        IBLANK=MAX0(35,WHERE)
C
C AND BLANK IT
C
        DO 15 I=1,IBLANK
15      LINE(I)=' '
C
C HMM... WHICH SIDE OF ZERO ARE WE ON?
C
        IF (WHERE .GT. 35)  THEN
C
C RIGHT SIDE
C
                         DO 20 I=36,WHERE
20                       LINE(I)='*'
                             ELSE
C
C LEFT SIDE
C
                         DO 30 I=WHERE,35
30                       LINE(I)='*'
                             ENDIF
C
C SET "ZERO"
C
        LINE(35)='+'
C
C AND THE SIN VALUE
C
        LINE(WHERE)='*'
C
C IF THIS VALUE IS < 35, SET SO WE OUTPUT TO ZERO LINE
C
        IF (WHERE .LE. 35)WHERE=35
C
C AND FINALLY OUTPUT THE LINE
C
        WRITE (6,21) (LINE(I),I=1,WHERE)
21      FORMAT (70A1)
100     CONTINUE
        CALL EXIT
        END
```

your computer book reviews

Tandy Colour

WITH its comprehensive content, and simple but methodical approach, *TRS80 Color Programs* (by Rom Rugg and Phil Feldman, published by Dilithium Press) is probably the next logical step after Tandy's excellent BASIC and Extended Colour BASIC manuals.

The book is compiled from material selected from magazines, and covers many areas of programming in BASIC, without too much emphasis on colour graphics. It is very easy for beginners to be carried away by visual effects, but this is well remedied.

Material on extended colour graphics is limited, and those graphics included re-teach many fundamentals about non-extended colour graphics. BASIC programming is covered by example, and specifically addresses areas including applications programs, educational programs and games.

All the programs cover various techniques, including files, arrays and data manipulation. None of them is particularly sophisticated or complicated, but the basic approach can be tailored for individual use. Each of the chapters discusses a particular program. First it gives a run-down on the basic function and how it is used, followed by a computer-generated listing, suggested changes, discussion of the main subroutines by line, listings of the variables used, and suggestions for adaptations.

The six sections of the book comprise: Applications of programs; Education programs of a fairly simple nature; Games programs; A cover of non-Extended Colour BASIC graphics with the emphasis on fundamentals; Mathematical programs; and Miscellaneous programs.

For the fairly methodical reader, who does not want to become overly involved, this style of writing has a lot to commend it. Basic descriptions are kept at a very understandable level, and the layout is pleasing. The book is not aimed at those interested in extended colour graphics, the "too clever" BASIC manipulations, and fancy PEEKs and POKEs. In essence, the authors have made a particular attempt at not trying to prove how clever they are, and have opted for a very readable style of writing. □

Doing Business With SuperCalc

HHHM, YES, WELL. The difficulty with *Doing Business With SuperCalc* (Sybex, Berkeley; 1983) is that it doesn't go far enough. The readers will break up into two groups: those who want more information on SuperCalc, and those who want more information on the kinds of business problems that can be solved by SuperCalc.

For the first group, the book stops short of the information in the SuperCalc manual. For example, there's no mention of the IF command in the reference section. On the other hand, there are commands in the book which won't work on my copy of SuperCalc – yet there's no mention of the version of SuperCalc used to develop the examples.

There's nothing in the book, from this point of view, that isn't in the manual, and the manual would seem to be more definitive about your particular copy of SuperCalc. Unless, of course, you don't have a SuperCalc manual – in which case, I'll bet I know how you acquired your copy of the program. Naughty, naughty.

From the point of view of discussing business problems which can be solved using SuperCalc, the book is a bit more satisfactory. Yet none of the examples demonstrates any novel techniques – or, indeed, anything that isn't in the manual.

Perhaps I can best liken *Doing Business With SuperCalc* to a book full of recipes for omelettes: one recipe for tomato omelette, one recipe for cheese omelette, one recipe for mushroom omelette, and so on. After reading the third recipe, I know that I should start by breaking an egg; reading it again and again is belabouring the point somewhat.

Likewise with this book. If reading lots of simple SuperCalc models helps you realise that SuperCalc works by rows and columns, great. However, if more time was spent examining the techniques and problems, rather than mechanics, the book would really be worthwhile. As it is, any budding executive who uses the key financial ratios model, without really understanding what is meant by Quick Ratio, Inventory Turnover and so on, deserves the trouble he's getting himself into.

These models smack to me of the old demagoguery: the computer produced this report, therefore it must be right. Of course, it's right as far as the input allows, but is it meaningful?

To sum up: heavy reading, of doubtful utility, except to owners of pirate copies, who will still have to fill in some blanks.

Our review copy was supplied by the publisher's Australian representative, the ANZ Book Company. □

Information Systems Design

I'VE FOUND *Information Systems Design* very useful, indeed, for providing overviews of techniques only otherwise mentioned in technical manuals.

Written by C P Brookes, P Grouse, D R Jeffery and M J Lawrence and published by Prentice-Hall Australia, the book is somewhat of a *tour de force*, an encyclopaedic treatment of a sweeping topic. Of necessity it is terse; presumably it was designed to fulfil the requirements of the information systems courses at the University of New South Wales, home of the authors. (Incidentally, readers may recognise Associate Professor Phillip Grouse as one of our Personal Computer of the Year selection panel.)

My library covers all kinds of personal computer hardware and software topics; *Information Systems Design* fills many of the gaps between them. In particular, it provides a professional-level treatment of many of the issues that microcomputer people are only beginning to realise are important, as well as techniques they have never heard of – which, incidentally, is why mainframe programmers are now starting to get the jump on the dyed-in-the-wool "I used to handle assembly code" microprocessor programmer.

The book ranges wide, over organisational issues, systems analysis and design, database management systems, communications, audits and security to decision support systems. Terse it may be, but it never completely loses touch

with commercial reality – for example, on page 349, Phillip Grouse does a neat sales pitch for DR PL/I-80.

So *Information Systems Design* has found its way into the reference section of my bookshelves, next to Gries' *Compiler Construction* (which I must read one of these days), and I keep turning to it for occasional reference. The first two chapters have been invaluable in teaching data processing to management students, as their summaries of key points make handy material for notes.

Commercial micro programmers could benefit greatly by browsing through this volume. □

Doing Business With Pascal

RICHARD AND Douglas Hergert have come up with a very interesting publication for those business people who have their own computers and aren't scared of them. In particular, small businesses with an Apple and Apple Pascal could benefit from *Doing Business With Pascal* (Sybex, Berkeley; 1983).

The book can be approached from two viewpoints: that of the businessman or programmer looking for information on how to write business programs, and that of the would-be Pascal programmer looking for sophisticated examples.

From the point of view of the businessman, the book provides one of the best overviews I have yet seen of a major corporate accounting system and management information system. In fact, it's so good that I've used it in my own teaching in preference over the "standard" text.

The discussion is sensible, not over-detailed, but delving into some interesting points. The early chapters form as good an introduction to business computing as I've seen recently.

From the programmer's point of view, the book represents a good second text on UCSD Pascal: I wouldn't recommend it as a first text, but its alternative viewpoints are just what is needed to hammer some points home. The explanations of program operation are lucid and clear, and reflect modern program design techniques. The approach throughout is orientated to meeting the user's needs.

I haven't tried the programs in *Doing Business With Pascal*, but I've dipped into most sections of it, and liked the approach I found. Don't buy it expecting it to be a cookbook accounting system *a la* the Osborne books – it's not – but those who are writing their own system will find much valuable information.

Our review copy was supplied by the publisher's Australian representative, the ANZ Book Company. □

Stringing Along

Ray Greet explains some important aspects of string-space management – allocation, string assignment and space recovery – in Microsoft BASIC. The information is based on extensive investigation using a Tandy TRS80 Model 1, but it is directly transportable to other implementations.

MICROSOFT, in its wisdom, chose to provide a dynamic system of string allocation for BASIC, rather than the pre-defined dimensioning of string variables that is required with some other implementations.

This philosophy, of course, is not without imperfections – what is? The system design is such that, periodically, a string-space recovery routine needs to be invoked whenever work space is exhausted.

The frequency and duration of invocation are very much dependent on user-program design and the string manipulation techniques used – delays due to recovery can range up to four minutes. Therefore, any technique that would reduce delays should be of value.

The allocation of string space occurs whenever BASIC is invoked (by whatever means) or whenever a "CLEAR n" statement is encountered. MBASIC (BASIC 80) is an exception, so I'm told: space allocation in this version is dynamic, but once allocated, all other conditions apply.

Three pointers are set in the communications area of memory to reflect the allocation: START, LIMIT and NEXT (my nomenclature). START points to the address for the beginning of string space, LIMIT points to the address of the end of string space and NEXT points to the address of the next byte available for assignment.

If the number of bytes specified by a "CLEAR n" statement is greater than that available, an "Out of Memory" error will result.

When BASIC is ready to move a generated string from its work area to string space, it tests to see whether the string can be accommodated between the bounds pointed to by NEXT and LIMIT.

If room exists, the string is moved; NEXT is updated to maintain "next byte" integrity; the string-pointer bytes of the appropriate variable are updated to point to the newly assigned string; and processing continues. (It can be that the string is not directly associated with a variable but is a byproduct of a string function – if this is the case, the address is loaded into the Literal String Pool Table.)

If the accommodation tests fails, then the recovery procedure is entered. Upon return, the assignment of the current string is re-attempted; if it is still unsuccessful, then an "Out of String Space" error occurs.

The function of string-space recovery is to gather all currently assigned strings, so that they are relocated to form contiguous bytes from the start of string space. After this recovery, bytes that are resident between the addresses pointed to by NEXT and LIMIT are released for further string operations.

The method is much more complex than just described. However, except for two very important points that follow, more detailed knowledge should not be required:

1. Before any one string can be relocated, all tables must be scanned to ascertain which string is currently located nearest to (but below and within string space) the current address contained in NEXT (NEXT is reset to START address when the routine is first entered).
2. Because of the way that routine is coded, any strings that may already be in contiguous bytes at the head of string space will be moved, byte by byte, to overwrite themselves!

From point one, analysis shows that the number of string variable tests required to complete a recovery is equal to the square of the number of string variables! Therefore the time taken can be reduced by reducing the number of string variables used. This means that local use of variables (that is, a variable used only for the current purpose) can be counter-productive.

From point two: if these strings could be protected, then they wouldn't have to be processed, though the associated variables are still tested. Despite this, benefits would still be gained as the variable string would fail an eligibility test.

Performing the following code will protect such strings:

```
1000 RM%=FNSI%(PEEK(START)
     +PEEK(START+1)*256
1010 IF FRE(" ")
1020 POKE START,PEEK(NEXT):
     POKE START+1,PEEK(NEXT+1)
```

Line 1000 reserves the current start location. Line 1010 forces execution of the string-space recovery routine. Line 1020 changes the address contained in START to that contained in NEXT.

This effectively protects strings that are located in memory above the address contained in NEXT. The protected strings can be accessed normally by their associated variables. Obvious for protection are any alphanumeric table data and/or constants loaded from on-line media. (The inclusion of this feature in a working application achieved a reduction of 25 seconds for string recovery).

Before exiting, this code should be executed:

```
1020 POKE START,RM%-INT(RM%/256)*256:
     POKE START+1,INT(RM%)/256
```

This restores START to pre-run status so that load errors are avoided.

Another prolific user of string space that can be corrected is the INKEY$ function: it consumes a byte of string space every time a key is struck. This can mean the execution of the recovery routine at more frequent intervals. When a key is struck under INKEY$ call, the ASCII code for the key is deposited in string space at the address indicated by NEXT.

This is in readiness for assignment to variable. By incorporating the following code, the key code struck can be extracted and deposited wherever the code (your code) dictates. This can, if used intelligently, effectively mean that INKEY$ will never use a byte of string space!

```
10 A$=""+" ":DEFFNSI%(A!)=-((`!>32767)
   *(A!-65536))-((A!<32768)*A!)
20 K%=FNSI%(PEEK(VARPTR(A$)+1)
   +PEEK(VARPTR(A$)+2)*256)
30 IF INKEY$="" THEN 30
40 POKE K%,PEEK(FNSE%(PEEK(NEXT)
   +PEEK(NEXT+1)*256))
```

Line 10 reserves a spot to store the input key code. It also defines a function which returns a signed integer that allows PEEKing and POKEing above 32767 decimal (Lewis Rosenfelder, *BASIC Faster and Better*). Line 20 initialises K% to the string address of A$. Line 30 waits for keyboard input. Line 40 gets input code from buffer and stores it in user variable.

Variations for the destination of the input are possible. However, the assignment must use the PEEK/POKE method; otherwise failure will result.

Of course, none of this is of much use unless the addresses of START, LIMIT or NEXT pointers are known. For the Tandy TRS80 Model 1, they are: START 40B1, LIMIT 40A0 and NEXT 40D6 (hex).

For other implementations, some research will be required if they're not documented; a means of inspecting memory, both user and system, is required. It will be even easier if the monitor has a search function.

Execute this program:

```
10CLEAR256:A$=""+"THIS IS A TEST"
```

This will reserve one page of string space and assign the string "THIS IS A TEST" to the variable A$ and locate the string in string space.

Use your monitor program to display memory in ASCII mode; you should have a memory map in your system documentation that shows the general area for string space. Search the memory for the sample string and, when found, note the addresses of the first and last letters "T", respectively. Decrease the first address by one – this gives NEXT content; the other address gives START content. Subtract (100 hex – 256 decimal – the value CLEARed) from START content, which will give LIMIT control.

As an example, take the situation of the 64 kilobytes (48-kilobyte user) used for research: the memory map shows that string space is allocated in high memory, immediately below any user-reserved memory. No memory has been reserved, so displaying the last memory page finds the string. The end of the string is located at address FFFE and the start at FFF1. Subtracting 100 (256 decimal) from FFFE gives LIMIT. The contents of START, NEXT and LIMIT respectively should be FFFE, FFF0 and FEFE.

```
F 4000 FEFF <enter>     40B1 returned
F 4000 F0FF <enter>     40D6 returned
F 4000 FEFE <enter>     40A0 returned
```

The manual for the TRS80 Model 1 shows that the "communications" area of memory commences at 400 hex; using the FIND function of the monitor used, I got 40B1, 40D6 and 40A0 – the required addresses, as reported earlier. Good hunting.

Another useful strategy is to force the invocation of the recovery routine (IF FRE(" ")) before entering a sub-function of a program – especially one that manipulates string data. The few seconds (if that) involved, which will probably go unnoticed, during this may preclude a recovery during execution of the function.

One final point: as recovery time is a dependency of the number of string variables assigned and not affected in actuality by the number of bytes allocated, it seems to me that as much space as possible should be CLEARed (bear in mind, of course, that space must be available for variable tables). This would further delay invocation of the recovery routine.

The string-handling operators of Microsoft BASIC allow complex string manipulations without the necessity of pre-definition of string fields (except any field that is to be used by LSET, RSET or MID$ =).

The string-space collection routine is a necessary part of the design philosophy. Its effect can be controlled, provided the programmer is prepared to optimise the code used, along with a mix of the recommendations herein. Programmers should dedicate some time toward gaining explicit knowledge of the effects each string operator – both singly and in conjunction with others – has on string space. □

Double the Density Double the Fun

Michael Hannon reviews the TRS80 Double-Density Kit, which he used to upgrade his Model 1 to a pseudo-Model 3.

AS I DON'T like paying out money for installation, and since Tandy Electronics planned to charge me extra for the privilege, I decided that I would install my newly acquired $229.95 Tandy TRS80 Double-Density Kit to my TRS80 Model 1.

It wasn't a problem. Despite the lack of instructions, it only took five minutes, in which time I opened the expansion interface and located the disk-controller chip – the largest one. This chip is removed from its socket (with considerable care), and the double-density board is then installed in the empty socket.

After that, you simply replace the disk-controller chip in the vacant socket and close up the expansion interface. There is no problem, as long as you take care to install the chips the correct way around, reducing the static electricity. However, being clever and doing the installation yourself, will void the warranty.

The manual gives the start-up sequence: turn on all peripherals and then the central processing unit (CPU), insert the double-density disk in drive 0 and press the reset button.

On loading, the screen clears and LOADING TRSDOS appears centred on the screen in 32-character format. Then the screen clears and a graphic picture of a computer comes up, *a la* Model 3. From then on, the humble TRS80 Model 1 acts just like a Model 3.

The first part of the manual describes how to make a back-up copy of the double-density disk. As I only have one disk, I had to do a single-drive copy. I carefully followed the instructions, which told me to write-protect the disk as a safety measure, but I found I couldn't back up – I received an error message when the disk operating system (DOS) told me to insert the source disk (the master disk).

I discovered that the master disk can't be write-protected to back up. After removing the write protect label, I was able to back up the master disk. When the destination disk has a flawed track during format, the computer hangs up while looking for the source disk.

After reading the manual, I proceeded to try to copy all my existing single-denstiy programs to double-density. As this disk-operating system supports single-drive copies and has a lot to say about loading in single-density disks and copying them to double-density, I thought there could be no problem.

Wrong again – even though the manual says nothing about single-drive copies in different density formats, I couldn't copy. A phone call to the Tandy hotline revealed that you can't copy different densities unless you have two disk drives.

I eventually copied all my machine-language programs by using a monitor program to load in the program under single-density and resetting the computer, loading the double-density disk, jumping to the monitor program and writing the file to disk (you could also use the DUMP command under TRSDOS, but it won't save programs under Hex 6FFF).

To copy BASIC programs, I loaded the program in under single-density, and then loaded the double-density disk and reset the BASIC pointers to the start of the program, and then saved it.

Apart from these problems, the rest of the software has been magnificent. The TRSDOS commands are almost the same as the Model 3 commands, including repeating keys, lower-case (with the lower-case kit installed) and a map of unused disk space. A DUAL command is provided to route output to the printer and screen. To change the number of tracks and stepping rate, you can configure the drive for different values, and even set the RS232 values using SETCOM.

There are 46 TRSDOS commands under double-density, as against 27 under single-density TRSDOS 2.3. Disk BASIC is another story – there are a few ways to start into BASIC, which includes setting memory size, number of files and automatically running your BASIC program from the TRSDOS command level.

Once in BASIC, you have 15 CMD commands, including a sort command, a compress program and a display-directory command. The single-density TRSDOS 2.3 has only five CMD functions.

Another very nice touch is the control of listing your program using the arrow keys, and pressing the comma key to edit the current line.

The rest of the disk BASIC functions are the same as single-density TRSDOS 2.3. To re-number a program, all you have to do is use the NAME command. And you don't have to load in BASICA as you do under TRSDOS 2.3.

In summary, apart from the problems in moving my software from single- to double-density, I'm very happy with the TRS80 Double-Density Kit. The version of disk operating system is more powerful than any previous TRSDOS for the Model 1, with most of the commands of the Model 3 implemented, along with a few commands which Model 3 doesn't have.

As well, the disk-storage space has increased from 58,880 bytes and only using 35 tracks, to 115,200 bytes with a one-drive system and BASIC on the disk, and using the full 40 tracks. The only disk drives that are unable to access 40 tracks are SHUGART drives. (They can only store 92,160 bytes with a one-drive system.) With a two-drive system, a formatted data disk can store 175,104 bytes on 40 tracks.

As far as I'm concerned, my $229.95 was money well spent, for I could have spent the same amount on more disks and another version of disk operating system, and still not have the storage space that double-density has. □

your computer SOFTWARE REVIEW

Superboard And Dabug — A Winner

By Garnett Znidaric

your computer

OWNER REPORT

I'M ALWAYS dubious of anything with a name prefixed by Super. And if you consider that many people are the same, and mix it with the modest approach Ohio Scientific puts into its advertising, you have a product that shouldn't sell regardless of how good it is.

But it is good...

At this point, I would like to introduce the Superboard single-board microcomputer. The 6502-based system takes the form of a single double-sided board which carries all chips and the keyboard. It uses "real" keys, in the QWERTY format.

When you add to this the on-board facilities for eight kilobytes of RAM, it looks like excellent value for around the $400 mark, plus power supply and monitor.

With eight kilobytes of BASIC in ROM, I found this machine quite easy to use and, with the help of a 6502 user group computer club, quite easy to expand to such things as a printer or disk system.
sive machines, but rare in this price bracket. They include such things as lower case characters, a high-res graphics generator, scientific functions and an auto repeat which functions simply by holding the desired key down for longer than half a second.

The Down Side

After spending 500 words or so explaining just how super the Superboard is, let me make a few comments about the drawbacks of Ohio Scientific.

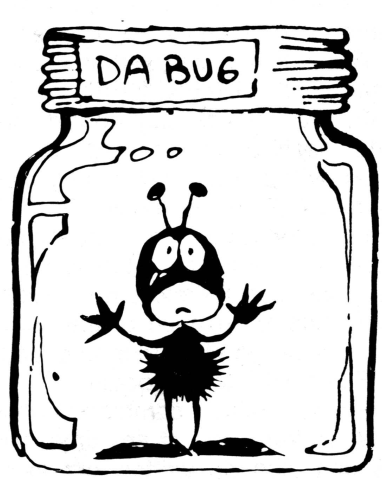
Apparently, in America, the system is rather well known in certain spheres and the parent company is seemingly happy with its sales. As a result, it doesn't really push its goods in overseas markets, such as Australia.

The single biggest fault with the Superboard is the lack of accurate documentation that accompanies the machine in its no-frills wrapper of Alfoil.

Being a no-frills machine, I suppose a good manual on programming is hard to manage, but on the other hand no one I know has yet been able to hook up a printer to the system using the instructions supplied — most have had to add at least one chip. *C'est la vie* (that's French for stiff...).

Apart from this, when I first received my micro there was no explanation on how to trim-in the video display and the cassette player. In essence this is very easy (there are two small trimmers on the top left corner of the board), as long as you know how...

The second annoying point is the standard screen size, 24 by 24. For most things, apart from games, this format is almost useless. Program lines look ridiculous, and word-processing is possible, but cramped.



Now let me introduce a bug that cures most.

Enter The Dabug III

Created by David Anear, the Dabug III is an EPROM which plugs into a ROM socket on the Superboard and introduces the following functions: full cursor control as well as line-copying facilities, single-key functions and, joy of joys, a 48-character wide screen.

Though my word-processor doesn't utilise the 48 by 12 format, programming is a breeze.

The main purpose of Dabug III is to speed up those time-consuming facets of programming, and this it quite ably does.

From the standard four kilobytes of RAM, the first expansion is to eight kilobytes. The second is the Dabug III. From then on, Ohio Scientific provides you with the key and the ticket, but it is up to you and your wallet just how far you journey.
requires a three-position, four-pole switch. In reality, it also needs a chip for the Clear To Send signal.

At this stage, the best system for expansion seems to be the Tasmanian Tasker system, which utilises the on-board disk bootstrap accessed through Ohio Scientific's 48-line bus. The Tasker system offers RAM up to 32 kilobytes, dual disk drives, PIA/VIA plus an Eprom board.

Soon to arrive will be a 64-character board, usable by my word-processing package. Then novels, as well as these short articles, will really start to flow.

The Final Analysis

For the beginner, the hobbyist, or even for a small business, I feel this is an excellent machine with unequalled versatility for the money.

As with most machines, just add a television set for a monitor, a tape recorder and a power pack, and you are on the air.

And be prepared to spend a few extra dollars on manuals and a membership in the nearest 6502 user group, such as KAOS or OMEGA . . . □

The Percom Connection

Some 18 months ago, when he changed to using disks on his System 80 microcomputer, Paul Wade bought a Micro-DOS system. Here, he reports on how it has stood the test of time.

UNLIKE most disk operating systems, the Micro-DOS system is fully resident in the computer memory, and doesn't require a system disk in drive 0. This tends to make it an excellent DOS for anybody with only one disk drive.

Micro-DOS is one of the two systems supplied by Dick Smith Electronics for use with its System 80 computers. The products also support the Tandy TRS80 Model 1, and instructions are included for changing the track and stepping rates of the DOS.

The memory required for Micro-DOS is seven kilobytes, a modest amount. As usual, it moves the BASIC pointers up, to allow it to sit low down in memory. The space used on the disk for the disk-operating system is 20 sectors – five percent of the disk – and, as such, it's feasible to initialise every disk with the disk-operating system for immediate booting.

The original disk comes with six utilities. Most are written in BASIC and lend themselves to user modification. Unlike other disk operating systems, it uses sequential file access; therefore, the disk is accessed at the sector level, rather than the granule level. (On a 40-track drive, there are 400 sectors, each with 255 bytes.) There is no directory sector, and the user is responsible for ensuring that no other programs are written over during disk I/O.

To help keep track of a disk's contents, the first utility is a disk-file manager, which uses 10 sectors to keep an encoded directory of the contents. However, it has to be updated by the user. A novel feature is while the file manager is running, simply entering the name of the program will cause the named program to be loaded and run automatically.

The second utility is mainly for disk operations. Format is, of course, essential (but it should be noted that CMD F 0, which does the same thing, may be entered at any time, even during the running of another program). Copy and back-up are also contained, the only difference being that the back-up does a format first, whereas the copy requests parts of the disk to be copied. FREE demonstrates the amount of space left on the disk at the sector level, ERASE clears sectors to empty status (this isn't essential, as it is possible to overwrite the sectors) and DUMP displays the sector contents as ASCII.

The third utility is a demonstration of how to make a notebook. This gives further information on the operation of Micro-DOS, but it can hold any information. Unfortunately, it doesn't allow keyed searches of its contents.

The fourth utility is a disk-diagnostic test (it is inherently destructive of data on the disk drive being tested; warnings are given by the program).

your computer

SOFTWARE REVIEW

The last two are special-purpose utilities. One is for applying patches to Micro-DOS itself, the other is for converting 80- to 96-track disks back to 40 tracks, either in single- or double-density. It is the only utility not written in BASIC.

Since Micro-DOS doesn't use its own imput-scanning routine, it relies on the reserved word-jumps, supplied by MicroSoft's Interpreter, with these words ignored: EOF, TIME$, OPEN, CLOSE, NAME, KILL. Fields may be defined allowing sectors to be encoded and packed, with normal disk I/O by way of GET and PUT at the sector level. The interrupt generated by the expansion interface is ignored, as there is no software clock, and therefore no reason for TIME$. Again, because of the sequential nature of the disk operating system, NAME and KILL aren't required.

Direct Micro-DOS commands allow a disk to be formatted at any time. Micro-DOS can also be written out to disk, while the autostart and initialising message can be changed even from within a program.

Normal LOADing and RUNning are supported from either command or execution modes. It should be noted that program-chaining isn't supported, leaving variables intact, but that a return to a menu program is easily accomplished.

There is also support for INSTRing searches, MID$ string replacement, hex constants, DEF FN and 10 DEF USR commands. All errors are reported in full by Micro-DOS and the program is left intact.

As a memory-dump utility isn't included, Percom has made a machine-code save utility, which is available on another disk, written in BASIC. The requirements for its use are that the program to be saved is resident in the memory, above the area used by Micro-DOS. The program requests the parameters of the program, writes a short pre-loader to disk, then dumps the memory to disk. The save utility is self-modifying, so it's unnecessary to modify the pre-loader.

One of the advantages of the Percom disk operating system is its user friendliness. It isn't as particular with the exact spacings of commands as other systems. For most users who are concentrating on BASIC or hybrid-BASIC programs, it should only take about half an hour to learn, including getting it to autostart directly from booting.

Saving of machine code will take a little longer, as it's necessary to provide a down loader to avoid conflict with the disk operating system. As usual, it's necessary to disable the disk-operating system before a system tape can be loaded – if you don't, a crash will result when the execution address is read.

On payment of appropriate royalties, Percom also allows Micro-DOS to be used as a transport vehicle for other programs, for which it has been well-designed.

The disadvantages of Micro-DOS are those associated with any program which requires separate data files to be kept – for example, Budget Manager by Tandy or Electric Pencil by Michael Shrayer. If the data file is to be kept from a BASIC program, then the program can be modified only with some difficulty. If the program is in machine code, the difficulty is probably more effort than it is worth. □

JOIN THE CLUB

AND GET MORE OUT OF THE VIC CENTRE CATALOGUE

★ Discounts*
★ Exchange Commodore Cartridge Service
★ Membership Specials

PAGE 22
CLUB
SPECIALS!
While Stocks Last.

MEMBERS

~~$149~~ – $129

~~$ 99.95~~ – $ 85

~~$ 56~~ – $ 47

NON MEMBERS

Purchase any **TWO** of the three items listed above at price listed in our Autumn '83 catlg. and you will receive a
FREE $30 MEMBERSHIP
to the VIC CENTRE COMPUTER CLUB

The VIC CENTRE
CW ELECTRONIC
416 Logan Rd., Stones Corner, Q'ld.
P.O. Box 274 Sunnybank Q'ld 4109,
Tel (07) 397 0888

★ CLUB MEMBER BONUS;
MEMBERSHIP DISCOUNT: For every one dollar spent with the VIC Centre you will receive a credit certificate for 12 cents against future purchases e.g. Commodore 64 $699 gives Credit Certificate of $83.88, etc. This does not apply to orders below $50, Club Specials, items purchased using Credit Certificates, or on trade-ins.

THE CLUB

The VIC CENTRE COMPUTER CLUB APPLICATION
CW Electronics 416 Logan Road, Stones Corner, Q'ld.
P.O. Box 274 Sunnybank Q'ld 4109

NAME: ....

ADDRESS: ....

....

SUBURB: .... P/CODE: ....

PHONE: ....

Membership Fee $55.00/$30.00

☐ Pay by Bankcard/
Bankcard No: ____________________

Expiry Date: __________

☐ Pay by Cheque/Money Order

Cartridge Title required: ____________________
(VIC-20 only at this stage)

Alternative ____________________

Signature:(Member and Card Holder) ....

RECEIPT & RULES

CONDITIONS OF MEMBERSHIP:

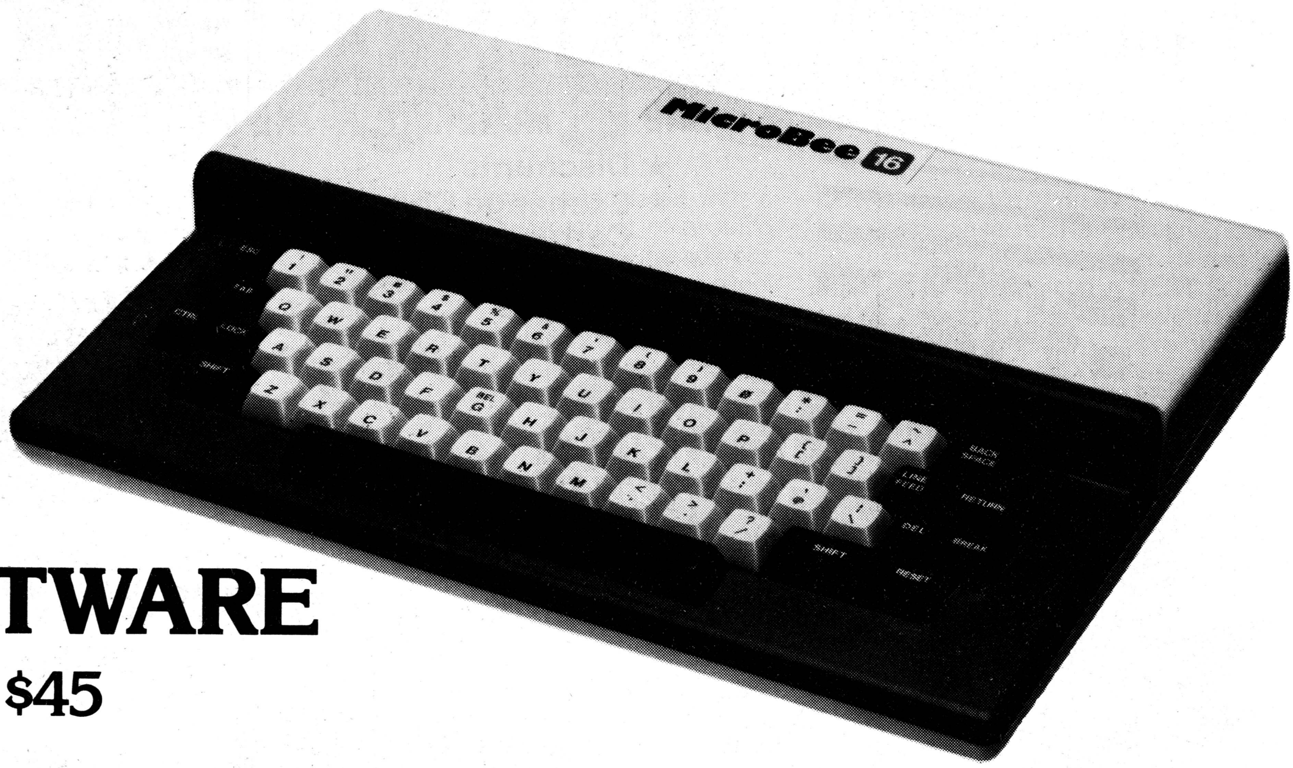
(1) Membership fee of $55.00 (incl. first Commodore cartridge) Non-cartridge member $30.00, both renewable annually. $10.
(2) Reduced membership rates are available with additional cartridges purchased at the time of making membership application.
(3) Exchange of Cartridges allowed at the rate of $7.00 at time of exchange. (+ $1 certified postage if applicable).
(4) The variety and number of cartridges available for exchange will be controlled by The VIC Centre.
(5) All Cartridges returned for exchange must be working and undamaged.
(5A) A service charge will be incurred on all repairable cartridges.
(6) Packaging and instructional material must be returned with the cartridge and in good condition.
(7) The cartridge returned must be the same cartridge borrowed.
(8) MAXIMUM time exchange is 1 calendar month, after this time the cartridge is unreturnable and deemed the property of the member.

Date: / /1983.

SIGNATURES:(Client) ....

(The VIC Centre) ....

NOTE: The signing of this form indicates agreement to the rules of membership.



SOFTWARE

Forth $45

A new language for the Microbee

This version was specifically written for the Microbee (not a converted Tandy or Sorcerer program) and supports sound and graphics. Execution speeds are only 30% slower than machine codes. Forth is self modifying and vastly speeds up program writing. This is a must if you want contact with a 'real' computer language . . . includes a comprehensive reference manual.

SPACE PATROL $16.95
High resolution real time game with sound and graphics. Similar to the popular 'scrambler' game.

UNDERWORLD OF KIN $14.95
Yes, this is one of ours! An adventure game in the classic tradition with graphics. Kill the wizard and escape from the underworld. Can you do it . . .?

GRAPH PILOT $14.95
Perfect for students. An excellent learning tool — specify the function and your microbee plots the graph.

BEE BUG $16.95
The perfect debugging tool combines extended monitor with disassembler/single step and a host of other features. This is worth a look.

SCREEN DUMP $14.95
Two machine code programs. Use them in your own programs. Produce both horizontal and vertical screen dumps to printer.

PENETRATOR $19.95
We have the rights to this popular game on the Microbee. Complete with editor to allow you to create your own game.

FINANCIAL DIARY $14.95
Sorry about the delay, but here it is. Calculate your monthly expenditure, profits, etc. Record appointments; the perfect time/money manager.

MONKEY MATHS $14.95
Great sound and graphics. Answer the question to stop the monkey pinching your coconuts. Makes it fun to learn.

MINE DROP $14.95
A fast high resolution game. Collect supplies with your tank and avoid the smart missile.

ADDRESS BOOK $14.95
Record names and addresses and recall them at the touch of a button. Simply enter the name and address or any other value and ADDRESS BOOK will find it for you.

BEE BASE II $19.95
The most powerful cassette based data base. Search/sort or do most anything you want. This is not a tutorial but a powerful tool. (Supplied with comprehensive manual.)

Coming Soon . . .

GRIDBLASTER. High res graphics game.
SPELLING GOLF. Have a round at golf, but only if you can spell.
BUSY GRAPH. A graph program for business people.

Do you want to sell your software (and make lots of money?)!
RING (02) 412 3838 or write to
P.O. BOX 793, CHATSWOOD 2067

DEALERS ENQUIRIES WELCOME

MicroBee Variables

By Eric B Lindsay

MicroBee owners don't yet have utility programs such as variable listers for programs, nor any method of locating variables in memory. These can be easily written if you understand precisely where and how the MicroBee stores variables, and this article explains how to locate and decode them.

MICROWORLD BASIC dedicates two bytes of memory as a pointer to the location in memory, of the value of each variable. It contains a variable table area from 1280 (500H) to 1696 (6A0H) or 416 bytes, in which are stored in strict alphanumeric order the 208 two-byte pointers for the real variables A0 to Z7.

There is another variable table area of 52 bytes from 1712 (6B0H) to 1764 (6B4H) for the pointers to the 26 integer variables A to Z, again in strict alphabetical order.

The pointers to any unused variable will contain 0. This makes it fairly easy to write a simple program in BASIC that goes through PEEKing at the variable table, and listing only those variables that are used. You can also PEEK locations from immediate mode.

You can use the monitor if you have Editor-Assembler or WordBee. However, as WordBee has a tendency to destroy BASIC programs, I would be cautious about using the monitor in WordBee to see what is stored in a program.

Each two bytes in the variable table print to the location of the start of the corresponding variable, and are in the usual low-byte, high-byte form. Those not familiar with hexadecimal arithmetic can find the location by multiplying the second (high) byte by 256, and adding the first (low) byte. This gives the location to PEEK for the start of the variable value.

The variables in a program are stored in the order in which they are encountered in the program, and – except for string variables – are stored immediately after the program in memory. BASIC takes care of where the program starts and ends, and where the next free space for numeric and string variables is, by means of another set of pointers which are located as follows:

Program start: 2256/7
Program end: 2258/9
Next available variable location: 2264/5
Next string variable location: 2268/9
Value to which SD is set: 2240

When you do a cold start, these pointers have values placed in them to ensure that BASIC programs start at 2304 (900H), and that the variable pointer is set to the next location in memory and the string pointer to 256 (100H) below the top of memory. Incidentally, you'll find these cold-start values stored in your BASIC 5.10 EPROMS, starting at BA3AH.

The values of the integer variables A to Z are stored as two bytes per variable, the same as pointers, in low-byte, high-byte form. You can find the value by multiplying the second byte by 256 and adding the first byte. You'll notice that the results of this give the value of the variable if it is between 0 and 32767. Negative numbers do not work so well.

In fact, at first glance, they don't appear to work at all. However, if you subtract the number you calculated from 65536, you'll come to the value of the variable. Negative numbers tend to be stored in computers in two's complement form, a complete description of which appears in most text books, or you can read Les Bell's article on binary and other arithmetics in the November and December 1981 issue of *Your Computer.*

Integer arrays are more complicated. When you dimension an array, say DIM A(2,3), you find that the array is actually three by four, since it starts from zero and goes to two, and from zero to three. Enough space is set aside in memory for each of the array variables stored two bytes each.

In memory, you'll find each array starting with 255, followed by the number of dimensions stored as a single byte. This is followed by a two-byte value for the size of each dimension except for the first one. The last two bytes are the size of the array in bytes. The actual values are stored after this, in the same form as regular integer variables.

Real variables A0 to Z7 are more interesting. The number of bytes used to store a real variable depends on the value you assigned to SD from BASIC. You require half the SD value in bytes, plus one byte. Since the normal setting for SD is 8, this means each variable occupies five bytes. If SD is four, you need three bytes; if SD is 14, you need eight bytes per real variable.

If this bit is on (if the value of the byte exceeds 128 or 80H), then the number is positive. If lower, the number is negative. The next lowest bit (bit six) is set on for positive exponents, and off for negative exponents. The remaining six bits of the first byte are the value of the exponent of the number, which explains the allowable range of the MicroBee's real variables – approximately plus and minus 64. The bytes that follow contain the actual number, two digits per byte, in binary coded decimal (BCD) form. I must admit I have never encountered BCD before in a home computer.

BCD is really easy to understand, if you happen to speak in binary or hexadecimal. Each byte contains eight bits, and these are taken four at a time, and used to code the 10 decimal digits zero to nine. The surplus bit patterns, which in hexadecimal would be indicated as A to F, are simply not used. This means that if you view the location of real variables using a monitor, which displays in hexadecimal, you will actually see the value of the variable displayed. However, life wasn't meant to be easy, so if you view these bytes as decimal numbers, they make little sense. You have to convert them to binary, take the top four bits and the bottom four bits separately, and convert these back to decimal. I told you that you needed to read Les Bell's articles on binary arithmetic...

Arrays of real variables are treated much the same as integer arrays. The first byte stored is 255, followed by the number of dimensions (the maximum allowed is 255, but limits of line length will prevent you getting anywhere near that limit) then the actual dimensions are stored, two bytes per dimension, in low-byte, high-byte form (except for the first dimension, which doesn't appear to be stored). After these is a two-byte count of the number of memory locations the array occupies. The values in each array variable come next, and these are stored in exactly the same manner as normal real variables.

The real fun comes in decoding string variables. The pointer in the real-vari-

ables pointer area points to one variable-storage area above the program. Here, you find zero, followed by a two-byte pointer in low-byte, high-byte format, followed by a count of the number of characters in the string, followed by more zeros to bring the total memory used to the same total as an ordinary real variable.

If you follow the pointer, you'll find the first string variable is stored 256 bytes below the top of memory, and that it is stored in reverse – that is, if you have a string "abcd", it's stored as "dcba".

Working backwards from the location at which the string is stored, near the top of the memory, you have the ASCII codes for the string, followed by 128, which indicates the end of the string, followed by yet another pointer. This points to the pointer that pointed to the string! At the end (or bottom, if you like) of all the string variables, you will find a 255.

That covers all the variables, as stored in the MicroBee. The program listed will find and display all variable storage areas, and also the program area.

As programs tend to occupy a lot of space, you may prefer to use it without lines 410 to 470. The results of the program are shown, with it displaying variables as used within itself. Lines 100 to 200 can be changed in any way, and merely include a fairly wide range of variables to give the program something to do. All sections of the program will run independent of the others, provided you include line 270. □

```
00100 REM DISPLAYS VARIABLES, TABLES AND PROGRAMS AS Nos 14/2/83
00110 LET A = 100
00120 LET B = -100
00130 LET C = A * B
00140 LET A0 = 100
00150 LET B7 = -100
00160 LET C0 = 1.2345678E08
00170 LET D7 = -1.2345678E07
00180 LET E0 = 9.8765432E-06
00190 LET F7 = -9.8765432E-05
00200 G0$ = "abcd"
00210 DIM G7(2,3)
00220 DIM H0(1,4)
00230 LET G7(0,0) = 4567.890
00240 LET G7(1,1) = 7654
00250 LET H0$(0,0) = "defg"
00260 LET H0$(1,4) = "xwvu"
00270 Z = 2256:J=0
00280 PRINT:PRINT "INTEGER VARIABLE POINTERS"
00290 FOR X = 1712 TO 1764
00300 IF J = 0 THEN PRINT: PRINT X;
00310 J = J + 1:IF J = 16 THEN LET J = 0
00320 PRINT PEEK(X);
00330 NEXT X
00340 PRINT:PRINT "REAL VARIABLES POINTERS"
00350 PRINT"Press any key when ready"
00360 IF KEY$ = ""THEN GOTO 360
00370 J=0:FOR X = 1280 TO 1696
00380 IF J = 0 THEN PRINT:PRINT X;
00390 J = J + 1:IF J = 16 THEN LET J = 0
00400  PRINT PEEK(X);: NEXT X
00410 PRINT:PRINT "PROGRAM AREA"
00420 J=0:PRINT  "Press any key when ready"
00430 IF KEY$ = "" THEN GOTO 430
00440 FOR X = PEEK(Z)+(256*PEEK(Z+1)) TO PEEK(Z+2)+(256*PEEK(Z+3))
00450 IF J = 0 THEN PRINT:PRINTX;
00460 J =J + 1:IF J = 16 THEN LET J = 0
00470 PRINT PEEK(X);:NEXT X
00480 PRINT:PRINT  "VARIABLES AREA"
00490 J=0:PRINT "Press any key when ready"
00500 IF KEY$ = "" THEN GOTO 500
00510 FOR X=PEEK(Z+2)+(256*PEEK(Z+3)) TO PEEK(Z+8)+(256*PEEK(Z+9))
00520 IF J = 0 THEN PRINT:PRINT X;
00530 J = J + 1:IF J = 16 THEN LET J = 0
00540 PRINT PEEK(X);: NEXT X
00550 PRINT:PRINT "STRING VARIABLES"
00560 J=0:PRINT "Press any key when ready"
00570 IF KEY$ = "" THEN GOTO 570
00580 FOR X = PEEK(Z+12)+(256*PEEK(Z+13)) TO 32512
00590 IF J = 0 THEN PRINT:PRINT X;
00600 J = J + 1:IF J = 16 THEN LET J = 0
00610 PRINT PEEK(X);
00620 NEXT X
00630 PRINT:PRINT "PROGRAM START";PEEK(Z)+(256*PEEK(Z+1))
00640 PRINT"PROGRAM END";PEEK(Z+2)+(256*PEEK(Z+3))
00650 PRINT"NEXT VARIABLE  LOCATION";PEEK(Z+8)+(256*PEEK(Z+9))
00660 PRINT"NEXT STRING VARIABLE LOCATION";PEEK(Z+12)+(256*PEEK(Z+13))
00670 PRINT"VALUE OF SD NOW SET TO ";PEEK(2240)
```

```
INTEGER VARIABLE POINTERS

 1712 15 15 17 15 19 15 0 0 0 0 0 0 0 0 0 0
 1728 0 0 180 15 0 0 0 0 0 0 0 0 0 0 0 0
 1744 0 0 0 0 0 0 0 0 0 0 0 0 0 0 182 15
 1760 0 0 178 15 0
REAL VARIABLES POINTERS
Press any key when ready

 1280 21 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1296 0 0 0 0 0 0 0 0 0 0 0 0 0 0 26 15
 1312 31 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1328 0 0 0 0 0 0 0 0 0 0 0 0 0 0 36 15
 1344 41 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1360 0 0 0 0 0 0 0 0 0 0 0 0 0 0 46 15
 1376 51 15 0 0 0 0 0 0 0 0 0 0 0 0 56 15
 1392 122 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1408 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1696 0

PROGRAM AREA
Press any key when ready

 2304 0 100 59 32 161 32 68 73 83 80 76 65 89 83 32 86
 2320 65 82 73 65 66 76 69 83 44 32 84 65 66 76 69 83
 2336 32 65 78 68 32 80 82 79 71 82 65 77 83 32 65 83
 2352 32 78 111 115 32 49 52 47 50 47 56 51 13 0 110 12

VARIABLES AREA
Press any key when ready

 3854 255 100 0 156 255 240 216 195 16 0 0 0 67 16 0 0
 3870 0 201 18 52 86 120 72 18 52 86 120 187 152 118 84 50
 3886 60 152 118 84 50 0 127 0 4 0 255 2 0 4 0 60
 3902 196 69 103 137 0 192 0 0 0 0 192 0 0 0 0 192
 3918 0 0 0 0 192 0 0 0 0 196 118 84 0 0 192 0
 3934 0 0 0 192 0 0 0 0 192 0 0 0 0 192 0 0
 3950 0 0 192 0 0 0 0 192 0 0 0 0 255 2 0 5
 3966 0 50 0 126 249 4 0 192 0 0 0 0 192 0 0 0
 3982 0 192 0 0 0 0 192 0 0 0 0 192 0 0 0 0
 3998 192 0 0 0 0 192 0 0 0 0 192 0 0 0 0 0
 4014 126 242 4 0 208 8 7 0 182 15 32

STRING VARIABLES
Press any key when ready

 32491 255 15 173 128 117 118 119 120 15 128 128 103 102 101 100 15
 32507 51 128 100 99 98 97
PROGRAM START 2304
PROGRAM END 3854
NEXT VARIABLE  LOCATION 4024
NEXT STRING VARIABLE LOCATION 32491
VALUE OF SD NOW SET TO  8
```

your computer text file

Osborne Users' Group

I WOULD like to inform your readers that there is now an official Osborne User Group. We publish a regular monthly newsletter and also have a library which is available to members and contains more than 80 different disks, full of games, utilities, applications programs and so on.

We are not controlled or influenced by the Osborne company in any way, but we do receive technical assistance and so forth.

For further details, please contact me by writing to the Ausborne User Group, PO Box C530, Clarence Street Post Office, Sydney, 2000, or by phoning (02) 81-1908 Sydney (after hours).

IAIN MacCULLOCH
Drummoyne, NSW

Perfect Defence

I MUST TAKE Les Bell to task for opinions expressed in the April issue, concerning the Perfect Writer program. His fundamental criticism is that Perfect Writer is not easy to use. It is a criticism I would like to answer by quoting his own writings from the April issue.

On page 48, we read that most word-processors use either a "diamond pattern" of keys or the arrow keys to control cursor movement, and a major problem with Perfect Writer is that it does neither. Perfect Writer, in its most common installation, the Kaypro, does, in fact, use the arrow keys. For really determined "diamondeers", the Perfect installation program provides the opportunity to create their own diamond to suit.

Quote: "Writers and typists do work with words, but when they think about cursor movement, they do so by visualising the direction of movement, not by thinking of the word which represents that direction. For this reason, I found the Perfect Writer scheme slow and awkward...." I respectfully suggest that the thing that separates writers and typists from mere computerists and others is that they instinctively know where each letter is on the keyboard. They do not all have a mental picture of a WordStar "diamond" firmly fixed in their mind. Every typist knows where "B for back" is on the keyboard. For them, there is no need to hunt for a cursor arrow, nor to learn and fixate on some new diamond image as surrogate arrows.

A further Les Bell quote, this time from page 36 and himself quoting another: "One with which I particularly agree is that 'the easiest software to use is not necessarily the best'. In other words, extra effort spent in learning to master a more powerful software package will generally be repaid." In recognition of these points, and because Perfect Writer, Perfect Calc, Perfect Speller and Perfect Filer all share the same command codes, I respectfully suggest that the "ease of use" rating in your Software Report Card should be upgraded from poor to very good.

When balanced against an equivalent package of WordStar, Mailmerge, VisiCalc, SpellStar and Cardbox, the Perfect package emerges as both easy to use and particularly good value for money. Of course, most of your readers who are likely to purchase this group of programs will be getting them "free" with their computer purchase.

Another Les Bell quote, from page 52: "I know you're all convinced by now that I'm some kind of prejudiced WordStar bigot." By their words shall we know them...

DAVID HILL
East Bentleigh, Vic

Wondai Group

QUEENSLAND'S Wondai Apple User's Group wishes to advise your readers that it is now distributing its newsletter, *Waug-Waug*, on disk to its members. The disk contains programs and articles.

Membership of the group is open to all Apple users; our current membership is 150.

Our address is PO Box 19, Wondai, 4606, or phone (071) 68-5606 Wondai (business hours).

DR P LIP
Wondai, Qld

Imported Profits

LIKE MANY others, I am about to take the plunge and purchase a home computer. Having waded through various glossy brochures, made phone calls to and visited computer shops, talked to computer owners and read magazines such as yours to gain knowledge and follow the trends, I have been able to devise a shortlist.

Three of the computers on my list were the MicroBee, the Commodore 64 and the BBC Micro. At that time, neither the 64 nor the BBC were available in Australia. However, from British magazines I knew their prices in Britain were 345 pounds and 399 pounds respectively, including VAT (value-added tax).

I note with considerable interest that both machines are now available here, at $699 and $1579 respectively. I believe both machines are imported from Britain and that VAT is not payable for export.

Allowing for customs duty and sales tax of 30 percent and an exchange rate of $1.76, then each machine – if purchased personally in Britain – could be landed in Australia for $684 and $776. I realise, of course, that there would be handling at both ends and shipping costs in addition for an importer to pay.

Do you think that the Australian buying public is being taken advantage of by the pricing whims of computer importers?

COLIN LOCK
Engadine, NSW

Moose's Free Ad

IN YOUR April edition, there was an article titled "Moose And His Motorcycle Mailing Lists". Well, I'm Moose.

I've formed a business called Surfway Software, located in the idyllic New South Wales South Coast village of Sussex Inlet. The company's prime objective is to develop business-applications software for the Apple microcomputer range.

Some of the systems so far developed and available include: Real-estate property management, trust accounting, general, creditors and debtors ledgers, payroll, poker-machine analysis, stock control/invoicing (soon to be released) and, as mentioned in your April article, a BMX club package and a motocross club package.

These systems are available as fully documented packages for the Apple II. Full price lists and systems summaries can be obtained by phoning me at (044) 41-2679 or writing to Surfway Software, PO Box 61, Sussex Inlet, 2540.

ANDY McVEIGH
Sussex Inlet, NSW

Colour Your World

WHOEVER sold Gerald Shea (*Your Computer*, January-February edition) a Tandy TRS80 Colour microcomputer without also mentioning *Australasian Rainbow* deserves a severe rap over the knuckles!

For the benefit of Gerard, and other readers with the same problem, *Australasian Rainbow* is a magazine dedicated to the TRS80 Colour computer and available on subscription from Greg Wilson, PO Box 9, Potts Point, 2011. The cost is $15 for six issues, and $29 for 12 issues.

In addition, Gerard should call his TRS80 Colour computer "network" contact, Eugene Staker (02 451-8060 Sydney) for details on local users' meetings. No one with a TRS80 Colour computer should suffer from lack of information.

ROSS HANSEN
The Gap, Qld

Nevada Cobol Groupies

YOUR READERS might be interested to learn of the formation of the Nevada Cobol Users' Group in the United States. This group was formed in January 1982 and has started publishing a monthly newsletter.

This newsletter is essential reading for anyone interested in getting some public-domain Nevada Cobol software or advice about Nevada Cobol.

The address is Nevada Cobol Users' Group, 5536 Colbert Trail, Norcross, Georgia 30092, United States.

MARK JAMES
Maroubra, NSW

your computer text file

MicroBee Searchers

WE HAVE a MicroBee and are interested to see if there is a users' group yet. As several letters have been published in *Your Computer* from MicroBee users who would, perhaps, like to start a club, we would be interested in linking up with them to form such a users' group.

Our MicroBee has an editor/assembler, 32 kilobytes of RAM and is hooked to a Dick Smith GP-80 printer. We have found it to compare well with other makes, except for a lack of software.

We can be contacted at 12 Highcliffe Close, Tullamarine, 3043. Our after-hours phone number is (03) 338-7363 Melbourne.

BILL AND MARK BELTON
Tullamarine, Vic

Heartache Territory

IT HAS BEEN said in many magazines that the Commodore VIC-20 is the world's best home computer. After buying one, I now know that this is not true.

The VIC-20 may be good for people who know a great deal about microcomputers, but for beginners such as myself it's just one heartache after another.

The user's manual which came with the computer isn't very helpful at all. The book is very confusing and far too complex. The three programs that are on the back few pages are very confusing and, for a beginner, they're too long and difficult.

Can anyone help me with some short and easy programs for my VIC-20? I can be contacted at 8 Ahern Street, Ayr, 4807.

JOHN JONES
Ayr, Qld

Sharp User Group

THE SHARP Computer Users' Association meets at Burwood Heights High School, Mahoney's Road, East Burwood (Victoria), on the second Friday of each month of the school year.

The association is a non-profit organisation. Our main interest at the moment is the PC1500/Tandy PC2, but the owner of any Sharp computer is welcome to attend meetings.

A I HARRIS
Syndal, Vic

Latrobe Valley Club

THE LATROBE Valley TRS80 Colour Computer User Group was formed at Morwell (Victoria) late last year, and began holding monthly meetings in January.

Anyone interested in the group can contact me at 31 Donald Street, Morwell, 3840.

GEORGE FRANCIS
Morwell, Vic

Peach Problems

WHEN YOU get your Hitachi Peach computer in this neck of the woods and discover the complete documentation (Peach's advertising) that you receive is a disk manual for L3 BASIC and you operate a cassette, and it keeps referring to a L3 BASIC manual, you really are up against it.

It's baffling that manufacturers spend thousands of dollars in magazine advertising, glossy brochures and so on and then fall down in explaining to the user – who is, after all, essential in the scheme of things – the elementary instructions and functions of the machine and what it is capable of.

N A WATTS
Busselton, WA

Perth Users' Group

I WOULD like to make Commodore VIC-20 owners aware of our VIC-Ups Computer Users' Group in Perth. Some of the benefits we are able to offer members include: help in solving problems; access to a library containing various books about the VIC-20, the BASIC language and computing in general; a very good range of software; courses in BASIC and machine code; eight-kilobyte memory boards; a four-slot mother board and other hardware items all of our own manufacture; and stimulating discussion about both hardware and software.

The group is a non-profit independent organisation not tied to any particular dealer or supplier, though we are able to arrange substantial discounts through a number of dealers.

Membership fees are $10 for a family group, plus a $5 nomination fee (full-time students are exempted from the nomination fee).

Anyone interested in joining should contact our secretary, Vic Smith, on (09) 390-7875 Perth, or myself on (09) 272-2018.

JIM PRIOR
Inglewood, WA

Date Confusion

I REFER TO Mr J Hastwell-Batten's recent article, "Updating dBase II", in which mention was made of the differing American and European methods of numerically identifying days, months and years.

I find the confusing use of numerals is easily avoidable by instead adopting Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. These abbreviations are easily understood in many other languages as well.

My comments are not to be interpreted to mean that I am in favour of abbreviations. On most documents, the date doesn't take up critical space – for example, the top right corner – and I cannot understand why it can't be spelled out in full, even though it may have been entered numerically or in an abbreviated form.

I'd be most impressed by software that would allow this.

J F MARSMAN
Lane Cove, NSW

Logical Dates

ONE OF THE main functions of computers these days is the storage and retrieval of information. The general rule in finding information from some source (encyclopedia or computer data bank) is to go from the general heading (for example, "computers") to the specific (say, "chips").

Logically, therefore, it is more appropriate to say "July 23rd" rather than "23rd July", as in most applications it is more appropriate to know what month we are talking about before we know the actual day.

Most digital watches are made in either Japan or the United States, and show the date in the "month/day" format. Surely it is less confusing to stick to this convention, rather than trying to enforce a 19th Century European convention on to a 20th Century technology? (*Shouldn't that be* Century 20th? **- Ed**).

If I ever get a program which shows the date in the American format, the only thing I'll be trying to do will be to change the number of the month for the name of the month (for example, July 23rd instead of 7/23).

ROGER COOPER
Woodford, NSW

Super-80 Review

MANY THANKS for publishing Robert Tanton's well-written review of the Dick Smith Super-80 computer kit in your January/February issue.

Basically, Mr Tanton seems to have given a very fair and objective evaluation of the Super-80 situation, at least as it existed around September last year. However, I would like the opportunity to advise your readers that the shortcomings to which he drew attention have since been overcome.

It did take us some time to provide worthwhile software, to be sure. However, there is now quite a lot of good software available: we have four software products on cassette, namely an editor assembler, a debug disassembler and two games. In addition, we have published a book of useful programs in BASIC, while another book of programs has been published by *Electronics Australia*.

We have also published several editions of our "Super-80 Users' Newsletter", which, judging from our correspondence, provides information and experiences. So we believe that software support for the kit is now very much better than it was – and compares very favourably with that available for any similar machine.

We agree with Mr Tanton that the original versions of the BASIC manual and the Monitor manual left a good deal to be desired. It was for this reason that we revised and expanded them quite considerably, as soon as staff resources permitted. As a result, we believe that they now give the user much more help than before.

In short, we are confident that with its greatly improved software support and documentation, the Super-80 now qualifies even more for Mr Tanton's description as "a good computer for you" if you're after "an inexpensive means to learn about programming and Z80 machine language."

JIM ROWE
Technical Director
Dick Smith Electronics
North Ryde, NSW

An Owner Defends...

I HAVE OWNED a Dick Smith Super-80 computer for 14 months and have been very impressed by it. In fact, I came away rather shocked after reading Robert Tanton's January/February article on the Super-80.

First of all, things have happened since the computer first appeared on the market. For instance, the Super-80 can now cater for floppy disks, a printer and a modem. This is achieved by purchasing an S-100 expansion board from Dick Smith Electronics at a cost of around $70. Apart from that, there are now five software companies, including Dick Smith Electronics, that cater for all types of software requirements. Software comes in the forms of games, utilities and education or if need be, special user support.

Bemak and Megasoft are two companies that I know which cater for the Super-80. Their addresses are: Bemak Pty Ltd, PO Box 218, Belconnen, 2616, and Megasoft, 151 Denton Avenue, St Alban's, 3021.

After learning about machine-language programming, through articles and books, I have discovered that the Super-80 is a very versatile machine.

GARRY BLACK
City Beach, WA

New ZX Service

A NEW SOFTWARE corporation has just been formed and, while it now only serves the Sinclair ZX81/Timex 1000 microcomputer systems, we hope to branch out to cover the Sinclair ZX80 and the Applied Technology MicroBee machines within the next few months.

Called Australian United Software, the corporation at present has only two branches. Ultimately, there will be a branch in every state, including Tasmania.

The services offered include a wide range of software, both on cassettes and as documented listings, and some hardware, as well as blank computer tapes at very low prices. All the software is original or is originally designed but based on other material, and comes from independent programmers.

Padular Programs has 16-kilobyte ZX81 programs on cassette, ZX81 hardware and blank tapes, while Rhiannon Software has a range of one-kilobyte ZX81 programs, selling either individually or as a collection package.

A catalogue of Padular sales is available free from Padular Programs, 22 Brand Street, Bundaberg, 4670 (but please send a stamped, self-addressed envelope) and a catalogue of Rhiannon programs is available for $2 (including postage) from Rhiannon Software, 72 May Street, West Preston, 3072.

NEVILLE PREDEBON
West Preston, Vic

Continued on Page 72

your computer tutorial

Understanding Assembler Part XII

Fresh from choosing the Personal Computer of the Year, then moving office (to escape the howling computer companies?), Les Bell takes up where he left off in our series...

WELL, DEAR READER, when last we spoke, we were discussing monitor programs, and had designed one with the rudimentary functions of dumping memory, changing memory and running programs. However, it wasn't much use for really debugging programs, as it lacked the ability to set and remove breakpoints.

Breakpoints are points in a program under development where you want to break out of the program and examine the state of memory and the processor registers. In assembly language, this is most easily done by replacing an instruction with a jump to the monitor, which then saves the processor registers and dumps them to the screen.

Because the shortest instruction in the 8080 set is only one byte long, we'll have to replace it with a one-byte jump instruction. Fortunately, such an instruction does exist: the **RSTn** instruction, which allows us to call one of eight locations in memory with only a one-byte instruction.

What our debugger must do, then, is store away the original instruction from the chosen breakpoint location, and replace

it with an RST instruction – in this case, RST 5. (If you're wondering why I chose RST 5, it's because SID already uses RST 7, and MP/M uses RST 6, so I couldn't use those – otherwise how would I debug the debugger?)

Here's the breakpoint setting routine itself. Nothing very tricky here:

```
*****************************************************************
*                                                               *
*                     breakpoint command                        *
*                                                               *
*****************************************************************

; set a breakpoint at an address given in command line

break:  call    spskip          ; skip over spaces
        call    getparm         ; get address of breakpoint
        mov     a,m             ; get the original instruction
        sta     instr           ; save it away
        mvi     m,0efh          ; replace it with a restart 5
        shld    tempad          ; save the breakpoint address
        ret
```

The RST 5 instruction will cause the program to, in effect, CALL location 0026H, and so we must ensure that our cold-start code places a JMP there to the debugger code, which saves the processor registers and prints them. In this case, I've called this routine TRAP. This little piece of code does that job:

```
mvi     a,(jmp)         ; jump for rst 5
sta     0028h
lxi     h,trap
shld    0029h
```

Now for the *piece de resistance:* the trap routine itself:

```
*****************************************************************
*                                                               *
*                       trap routine                            *
*                                                               *
*****************************************************************

; encountering a breakpoint sends the processor here to
; print the contents of the registers
```

The first thing the routine does is to exchange HL with the top of the stack. This does two things: first, it saves HL on the stack; secondly, it fetches the return address of the RST 5 instruction into HL (remember, RST 5 is really a CALL which places the return address on the stack).

Now, the return address is the address *after* the breakpoint, so we must decrement the program counter before storing it away. Next, we push the remaining registers on the stack, so that they are safely beyond harm's reach on the stack:

```
trap:
xthl                    ; get breakpoint address
dcx     h               ; pc is one too high
shld    tempad          ; save the breakpoint address
push    d
push    b
push    psw
```

Having done this, we can load HL with the value of the stack pointer and use it to read the register values off the stack before printing them:

```
lxi     h,0
dad     sp
```

Then we get the first byte off the stack. This byte is half the program status word, and contains the flags:

```
mov     a,m             ; A contains flags
call    flprt           ; print them
```

The easiest way to display the flags is with a general purpose routine which displays any byte in binary:

```
flprt:  mvi     d,8             ; set counter to number of bits
fl1:    ral
        mov     b,a             ; save flags in B
        mvi     a,'0'           ; get an ASCII zero
        aci     0               ; and if carry is set, make '1'
        mov     c,a             ; then put it in C
        call    putch
        mov     a,b             ; retrieve flags
        dcr     d               ; count down
        jnz     fl1             ; and loop again
        ret
```

The only real trick in this routine is the use of the carry bit to decide whether to print 1 or 0; I've never seen this technique used elsewhere, but I'm sure someone else must have thought of it.

The next task is to print the contents of the accumulator, which is the next byte on the stack. It might be a good idea, from this point on, to label what we print across the screen, so I included short messages in the code and wrote a simple in-line print routine to handle them:

```
ilprt:                          ; in-line print subroutine
        xthl                    ; get ptr and save in HL
ill:    mov     a,m             ; get char
        ora     a               ; reached end?
        jz      ilex            ; yes, exit
        mov     c,a             ; move into C
        call    putch           ; and print
        inx     h               ; point to next
        jmp     ill             ; and go round
ilex:   inx     h               ; point to byte after end 0
        xthl                    ; restore HL
        ret                     ; and return
```

Notice that when this routine is entered, the top of the stack is the return address, which (not entirely by accident) is also the address of the first byte of the message to be printed. This byte is checked to see if it's zero (the string terminator), then output; and so we continue through the string. Finally, HL points beyond the string, and we stick it back on the stack and do a return – to the first instruction after the string.

This technique, therefore, requires the message to be written in the code, immediately after the "call ilprt" instruction:

```
call    ilprt           ; print message
db      ' A= ',0
inx     h               ; point to 'A' on stack
mov     a,m             ; get the value
call    h8              ; and print it
```

And so it continues, now with the remaining register pairs. Notice that the 8080 places 16-bit values in memory with the two bytes in reverse order. So I wrote a dead-simple routine to get the two bytes in reverse order and print them:

```
call    ilprt
db      ' BC= ',0       ; print BC
call    trl             ; print register pair from memory
call    ilprt
db      ' DE= ',0       ; ditto DE
call    trl
call    ilprt
db      ' HL= ',0       ; ditto HL
call    trl
```

Finally, we know that we have saved four register pairs since encountering the RST 5 instruction. Therefore, by adding to the current value of the stack pointer, we should (and do) have the value of SP just before the breakpoint:

```
call    ilprt
db      ' SP= ',0
lxi     h,8             ; number of registers saved
dad     sp              ; that's original stack value
call    h16             ; print it
```

We also know the value of the breakpoint, as it was stored in tempad as we entered the breakpoint routine. So we print it, and also save the address ready to resume – but it must be saved in a new address in case we set another breakpoint.

Finally we restore the instruction which the breakpoint had displaced, and jump to the main monitor routine:

```
        call    ilprt
        db      ' PC= ',0
        lhld    tempad          ;
        call    h16
        lhld    tempad          ; get breakpoint address
        shld    lastbrk         ; save for resume
        lda     instr           ; get instruction
        mov     m,a             ; and restore it
        jmp     monl            ; go to mainline

trl:    inx     h               ; increment past E 'cos
        inx     h               ; DE reversed on stack
        mov     a,m             ; get D
        call    h8
        dcx     h               ; point to E
        mov     a,m             ; get it
        call    h8
        inx     h               ; increment past D
        ret
```

The resume command allows us to continue execution after a breakpoint. It simply restores the registers by popping them in reverse order. Bear in mind that the scanner called this routine, so there is an extra return address on the stack which we must get rid of first:

```
**********************************************************************
*                                                                    *
*                         resume command                             *
*                                                                    *
**********************************************************************

resume:                         ; resume operation after break
        pop     psw             ; pop extra return address
        pop     psw
        pop     b
        pop     d
        lhld    lastbrk         ; get breakpoint address
        xthl                    ; get return address
        ret                     ; and go there
```

The only thing left to do is to allocate some storage for the various variables we have used; this should be placed at the end of the program:

```
instr   ds      1
tempad  ds      2
lastbrk ds      2
```

There are several improvements that could be made to this program. The first glaring omission is that the debugger makes no attempt to maintain a separate stack for its own use; it simply sticks registers and its own internal return addresses on to the stack of the program it is debugging. While this is all right if that program is using the debugger's stack, which allows 32 levels of pushing, you should be aware whether or not your programs maintain their own stack.

It is possible to rewrite the trap and resume routines to switch stacks, and this is left as an exercise for the reader (my way of saying, "Why should I do all the work?").

The next major improvement would be to add single-stepping, which is done by continually inserting breakpoints. The major difficulty here is that the 8080 instructions vary in length - one, two or three bytes – and the debugger must know how long each is, in order to place a breakpoint after it. Nonetheless, it can be done (how do you think DDT works?).

Meanwhile, bear in mind that it's possible to write your program with multiple RST 5's already in the code for debugging purposes. Once the program is debugged, you can take out those restarts.

Another improvement would be to tidy up the flag-printing routine to label each flag, or do what SID does: print dashes for reset flags, and initials for the ones that are set.

What happens if you resume before encountering a breakpoint? Perhaps it might be a good idea to prevent that, and while you're at it, extend the idea so that you can only resume once after a breakpoint.

That really wraps up all the elementary features of a monitor program. In designing this program, we have used a number of techniques and programming tricks. The code has been fairly modular, so the program contains a number of subroutines which may be helpful to you in your own programming.

In the next article, I'll move on to start on file input/output under CP/M, using the construction of a word-counting program as an example. Meanwhile, here's the completed monitor program:

```
; 8080 / Z-80 monitor routines
;

false   equ     0
true    equ     not false

rmac    equ     false

ctrc    equ     03h                     ; control-C for abort
acr     equ     0dh                     ; carriage return
alf     equ     0ah                     ; line feed
ctrh    equ     08h                     ; backspace character
tab     equ     09h
ctrs    equ     13h                     ; control-S for pause output
ctrx    equ     18h                     ; control-X erases input line
clear   equ     1ah                     ; TVI 910 clear screen cmd

del     equ     7fh

prompt  equ     '>'
buflen  equ     128
exit    equ     0000h

        title   'monitor rev 1.4'

        if      not rmac
        org     0100h
        endif

monitor:
        jmp     boot
monl:   call    crlf
        mvi     c,prompt
        call    putch
        lxi     h,buff
        call    getln
        lda     buff
        cpi     acr
        cnz     scanner
        jmp     monl

getln:  mvi     e,0             ; character counter
getlnl: call    conin           ; get a character
        cpi     ' '             ; is it a control char?
        jc      getln4          ; yes, jump to handler
        cpi     'Z'+1           ; is it lower case?
        jc      getln2          ; no, carry on regardless
        ani     5fh             ; otherwise fix it
getln2: mov     m,a             ; store char in buffer
        mvi     a,buflen        ; get buffer length
        cmp     e               ; have we reached it?
        rz                      ; yes, return to caller
        mov     a,m             ; retrieve character
        inx     h               ; bump buffer pointer
        inr     e               ; and counter
        mov     c,a
        call    putch           ; echo char
        jmp     getlnl

getln4: cpi     ctrh
        jz      getln5
        cpi     ctrx
        jz      clline
        cpi     acr
        jnz     getlnl
        mov     m,a

crlf:   mvi     c,acr
        call    putch
        mvi     c,alf
        call    putch
        ret
```

```
getln5:                         ; control-H (backspace) handler
        mov     a,e
        ora     a
        jz      getln1
        mvi     c,ctrh          ; print backspace
        call    conout
        mvi     c,' '           ; then a space
        call    conout
        mvi     c,ctrh          ; then another backspace
        call    conout
        dcr     e               ; count down
        dcx     h               ; back up buffer pointer
        jmp     getln1          ; get next character

clline:                         ; control-X (clear line) handler
        mvi     c,ctrh          ; print backspace
clline1:
        mov     a,e
        ora     a
        jz      getln
        call    conout
        mvi     c,' '           ; print space
        call    conout
        mvi     c,ctrh          ; print backspace
        call    conout
        dcx     h               ; back up buffer pointer,
        dcr     e               ; count back a char
        jnz     clline1         ; to beginning of line
        jmp     getln           ; and start all over
putch:
        call    conout
        call    const           ; test for input character
        ora     a               ; set flags to test for zero
        rz                      ; no character, so return
        call    conin
        cpi     ctrs            ; control-S?
        jz      pawz            ; yes, pause
        jmp     mon1            ; no, halt
pawz:   call    conin
        cpi     ctrc            ; control-C?
        jz      mon1            ; yes, abort
        ret

        page
*****************************************************************
*                                                               *
*                       simple scanner                          *
*                                                               *
*****************************************************************

scanner:
        lxi     h,buff
        mov     a,m             ; get a character
        inx     h               ; point to next
        shld    cursor          ; and save cursor
        sui     'A'             ; convert to table index
        jc      error           ; if less than A then error
        cpi     'Z' - 'A' + 1   ; if > 'Z' then error
        jnc     error
        add     a               ; double A
        lxi     h,table         ; point to start of table
        mvi     d,0
        mov     e,a             ; put offset in DE
        dad     d               ; and add it to HL
        mov     e,m             ; get low byte
        inx     h
        mov     d,m             ; get high byte
        xchg                    ; get it into HL
        pchl

table:
        dw      error           ; A
        dw      break           ; B
        dw      error           ; C
        dw      dump            ; D
        dw      exit            ; E
        dw      error           ; F
        dw      go              ; G
        dw      error           ; H
        dw      error           ; I
        dw      error           ; J
        dw      error           ; K
        dw      error           ; L
        dw      error           ; M
        dw      error           ; N
        dw      error           ; O

        dw      error           ; P
        dw      error           ; Q
        dw      resume          ; R
        dw      subst           ; S
        dw      error           ; T
        dw      error           ; U
        dw      error           ; V
        dw      error           ; W
        dw      error           ; X
        dw      error           ; Y
        dw      error           ; Z

*****************************************************************
*                                                               *
*                       error handler                           *
*                                                               *
*****************************************************************

error:
        mvi     c,'?'
        call    conout
        call    crlf
        jmp     wboot

        page
*****************************************************************
*                                                               *
*                       scanner tools                           *
*                                                               *
*****************************************************************

spskip:                         ; skip over spaces and delimiters
        lhld    cursor
spsk1:  mov     a,m
        inx     h
        cpi     acr             ; no cr expected, error
        jz      error
        cpi     ' '
        jz      spsk1
        cpi     ','
        jz      spsk1
        cpi     tab
        jz      spsk1
        dcx     h
        shld    cursor
        ret

getparm:                        ; get hex parameter
        push    d
        push    b
        lxi     h,0
gp1:    call    getchar
        call    gp2
        jc      gp3
        dad     h
        dad     h
        dad     h
        dad     h
        ora     l
        mov     l,a
        jmp     gp1

gp2:

; convert ASCII to binary, return with carry set if not a valid digit.

        sui     '0'             ; remove ASCII offset
        rc                      ;  character < 0
        cpi     'F'-'0'+1
        cmc                     ; complement carry
        rc                      ; character > F
        cpi     10
        cmc                     ; complement carry
        rnc                     ; ok, number 0 - 9
        sui     'A'-'9'-1
        ret

gp3:                            ; character not hex
        call    ungetch
        pop     b
        pop     d
        ret

getchar:                        ; get character from buffer
        push    h               ; save HL
        lhld    cursor          ; get cursor
        mov     a,m             ; get character
        inx     h               ; point to next
        shld    cursor          ; save cursor
        pop     h               ; restore HL
        ret

ungetch:                        ; back up cursor
        push    h               ; save HL
        lhld    cursor          ; get cursor
```

```
        dcx     h               ; move back
        shld    cursor          ; save cursor
        pop     h               ; restore HL
        ret

        page

*********************************************************************
*                                                                   *
*                              dump                                 *
*                                                                   *
*********************************************************************

dump:
        call    spskip
        call    getparm         ; get start address
        xchg                    ; put start into DE
        call    spskip
        call    getparm         ; get finish
        xchg                    ; DE <- finish, HL <- start
d0:     push    h               ; save base pointer on stack
        call    hl6             ; print initial address
        mvi     c,tab           ; and tab
        call    putch
d1:     mov     a,m             ; get byte from memory
        call    h8
        mvi     c,' '           ; print a space
        call    putch
        inx     h               ; point to next byte
        call    d8              ; have we reached the end?
        jc      d2              ; dump remaining ascii
        mov     a,l
        ani     0fh             ; mask lower bits
        jnz     d1              ; if not zero, keep dumping
d2:     mvi     c,' '           ; else space and keep dumping
        call    putch
        pop     h               ; get base pointer
d4:     mov     a,m             ; get char from memory
        ani     7fh             ; strip msb
        cpi     ' '             ; if less than space
        cm      d7              ; replace with a dot
        mov     c,a             ; move into C
        call    putch           ; and print it
        inx     h               ; point to next
        call    d8
        jc      crlf            ; exit with a CRLF
d5:     mov     a,l
        ani     0fh
        jnz     d4
        call    crlf
        jmp     d0

d7:     mvi     a,'.'
        ret

d8:                             ; reached end yet?
        mov     a,e
        sub     l
        mov     a,d
        sbb     h
        ret

        page

*********************************************************************
*                                                                   *
*                     S (substitute) command                        *
*                                                                   *
*********************************************************************

; display memory with option of changing, byte by byte

subst:
        call    spskip          ; skip over spaces
        call    getparm         ; get start address
subst1:
        call    hl6             ; output address
        mvi     c,' '           ; and then a space
        call    putch
        push    h               ; save address on stack
        mov     a,m             ; get memory contents
        call    h8              ; output byte in A
        mvi     c,' '           ; space in C
        call    putch           ; and output it
        lxi     h,buff          ; point to buffer
        call    getln           ; get a hex number (or whatever)
        lxi     h,buff          ; point to beginning of buffer
        shld    cursor          ; set cursor there
        mov     a,m             ; get first char
        cpi     '.'             ; is it a dot?
        jz      mon1            ; back to mainline
```

```
        cpi     acr             ; if it's a cr, don't change
        jz      subst2          ; skip over change code
        call    getparm         ; get hex number
        mov     a,l             ; mov byte into a
        pop     h               ; get back address
        mov     m,a             ; and store byte
        inx     h               ; move on to next
        jmp     subst1          ; and loop round
subst2:
        pop     h               ; restore stack
        inx     h               ; move on
        jmp     subst1          ; and loop round

        page

*********************************************************************
*                                                                   *
*                            go command                             *
*                                                                   *
*********************************************************************

; go (jump) to an address from command line

go:     call    spskip          ; skip over spaces
        call    getparm         ; get address to go to
        xchg                    ; save in DE
        lxi     h,wboot         ; get return address
        push    h               ; and place it on stack
        xchg                    ; get go address again
        pchl                    ; and go

*********************************************************************
*                                                                   *
*                        breakpoint command                         *
*                                                                   *
*********************************************************************

; set a breakpoint at an address given in command line

break:  call    spskip          ; skip over spaces
        call    getparm         ; get address of breakpoint
        mov     a,m             ; get the original instruction
        sta     instr           ; save it away
        mvi     m,0efh          ; replace it with a restart 5
        shld    tempad          ; save the breakpoint address
        ret

        page

*********************************************************************
*                                                                   *
*                           trap routine                            *
*                                                                   *
*********************************************************************

; encountering a breakpoint sends the processor here to
; print the contents of the registers

trap:
        xthl                    ; get breakpoint address
        dcx     h               ; pc is one too high
        shld    tempad          ; save the breakpoint address
        push    d
        push    b
        push    psw
        lxi     h,0
        dad     sp
        mov     a,m             ; A contains flags
        call    flprt           ; print them
        call    ilprt           ; print message
        db      ' A= ',0
        inx     h               ; point to 'A' on stack
        mov     a,m             ; get the value
        call    h8              ; and print it
        call    ilprt
        db      ' BC= ',0       ; print BC
        call    trl             ; print register pair from memory
        call    ilprt
        db      ' DE= ',0       ; ditto DE
        call    trl
        call    ilprt
        db      ' HL= ',0       ; ditto HL
        call    trl
        call    ilprt
        db      ' SP= ',0
        lxi     h,8             ; number of registers saved
        dad     sp              ; that's original stack value
        call    hl6             ; print it
        call    ilprt

        db      ' PC= ',0
```

```
        lhld    tempad
        call    h16
        lhld    tempad          ; get breakpoint address
        shld    lastbrk         ; save for resume
        lda     instr           ; get instruction
        mov     m,a             ; and restore it
        jmp     monl            ; go to mainline

trl:    inx     h               ; increment past E 'cos
        inx     h               ; DE reversed on stack
        mov     a,m             ; get D
        call    h8
        dcx     h               ; point to E
        mov     a,m             ; get it
        call    h8
        inx     h               ; increment past D
        ret

flprt:  mvi     d,8             ; set counter to number of bits
fll:    ral
        mov     b,a             ; save flags in B
        mvi     a,'Ø'           ; get an ASCII zero
        aci     Ø               ; and if carry is set, make '1'
        mov     c,a             ; then put it in C
        call    putch
        mov     a,b             ; retrieve flags
        dcr     d               ; count down
        jnz     fll             ; and loop again
        ret

ilprt:                          ; in-line print subroutine
        xthl                    ; get ptr and save in HL
ill:    mov     a,m             ; get char
        ora     a               ; reached end?
        jz      ilex            ; yes, exit
        mov     c,a             ; move into C
        call    putch           ; and print
        inx     h               ; point to next
        jmp     ill             ; and go round
ilex:   inx     h               ; point to byte after end Ø
        xthl                    ; restore HL
        ret                     ; and return

*******************************************************************
*                                                                 *
*                       resume command                            *
*                                                                 *
*******************************************************************

resume:                         ; resume operation after break
        pop     psw             ; pop extra return address
        pop     psw
        pop     b
        pop     d
        lhld    lastbrk         ; get breakpoint address
        xthl                    ; get return address
        ret                     ; and go there

        page

*******************************************************************
*                                                                 *
*                    hex output routines                          *
*                                                                 *
*******************************************************************

h16:
        mov     a,h
        call    h8
        mov     a,l
h8:
        push    psw
        rrc
        rrc
        rrc
        rrc
        call    h4
        pop     psw
h4:
        ani     Øfh
        adi     'Ø'
        cpi     '9'+1
        cp      hadj
        mov     c,a
        call    putch
        ret
hadj:   adi     'A'-1-'9'
        ret

        page
```



```
*******************************************************************
*                                                                 *
*                         fake bios                               *
*                                                                 *
*******************************************************************

data    equ     Øe3f8h          ;data port
stat    equ     Øe3f9h          ;status port
rbf     equ     ØØØØØ1ØØb       ;receive buffer full bit
tbe     equ     ØØØØ1ØØØb       ;transmit buffer empty bit

pio     equ     2Øh             ; parallel I/O chip (8255A)
lstat   equ     pio+2
ldata   equ     pio+1
pcont   equ     pio+3

; JUMP TABLE
fbios:  jmp     boot
        jmp     wboot
        jmp     const
        jmp     conin
        jmp     conout
        jmp     list
        jmp     punch
        jmp     reader

boot:   mvi     a,1Ø1Ø11ØØb     ; set 8255 mode
        out     pcont
        mvi     a,ØØØØØ1Ø1b     ; set group b mode
        out     lstat
        mvi     a,(jmp)         ; jump for rst 5
        sta     ØØ28h
        lxi     h,trap
        shld    ØØ29h
        mvi     c,clear
        call    conout

wboot:  lxi     sp,stk
        jmp     monl

; console status routine, returns Ø if no char, ØFFH if character avail.

const:  lda     stat
        ani     rbf             ; mask rbf bit
        mvi     a,Ø             ; if no data, return with zero in A
        rnz
        dcr     a               ; otherwise put Øffh in A
        ret                     ; and return

; console input routine

conin:  lda     stat            ; get status from UART
        ani     rbf             ; mask rbf bit
        jnz     conin           ; wait for character
        lda     data            ; get character
        cma                     ; adjust for negative bus
        ani     7fh             ; strip high bit
        ret

; console output routine

conout: lda     stat            ; get status from UART
        ani     tbe             ; mask tbe bit
        jnz     conout          ; wait for buffer to empty
        mov     a,c             ; move character into A
        cma                     ; adjust for negative bus
        sta     data            ; and send it
        cma                     ; make positive again
        ret

list:
        in      lstat           ; get lp status
        ani     2
        jz      list
        mov     a,c
        out     ldata
        ret

; other functions dummies for this example

punch:
reader:
        ret                     ; return

buff    ds      128
        ds      64
stk     equ     $
cursor  ds      2
instr   ds      1
tempad  ds      2
lastbrk ds      2
        end
```

Bug Blast	$19.95
Snakeout	$17.95
Cattle Round Up	$16.95
Head On	$17.95
Paratrooper	$19.95
Cricket	$16.95
Mow	$16.95
Bombs Away	$16.95
Space Pak	$16.95
Artillery	$16.95
Alien Invasion	$19.95
Adventure Pak (8k, 16k)	$19.95
Target Command	$16.95
Football Challenge (8k)	$19.95
Dungeons of Kal	$19.95
Convoy Escort	$19.95
Convoy Raider	$19.95
Boxers Corner	$19.95
Computer Baseball	$17.95
Star Defender	$19.95
At The Track	$19.95
Adventure Pack 1	$19.95
Adventure Pack 2	$19.95
Grave Robbers	$19.95
Victory Casino	$14.95
Get Four	$15.95
Night Rider	$15.95
Cosmic Debris	$16.95
Games Pack 1	$16.95
Games Pack 2	$16.95
Ludwig's Lemon Lasers	$16.95
Masterword	$19.95
Mancala	$12.95
Kongo Kong	$19.95
Annihilator	$19.95
Trek	$16.95
Street Sweepers	$15.95
Allied Defence	$19.95
Mail it...20	$19.95
Assembler Editor	$19.95
Blasteroids	$16.95
Cosmic Crusader	$19.95
Ultimate Tank	$19.95
Chimp Chase	$16.95
Checkbook (3,8,16k)	$24.95
Tunnel Patrol	$14.95
Vic Vango	$14.95
Fruitflys	$14.95
Galactic Crossfire	$19.95
Horsefile (3,8,16k)	$24.95
Dungeon Quest (16k)	$19.95
4 in 1 Games Special	$19.95
3 Slot Expansion Board	$59.95
Copycalc	$19.95
Vic Twist	$16.95
Alphabet Soup	$17.95
Vic Trip & Party Planner	$16.95
Cribbage	$16.95
Grandell Island	$19.95
Vic Amortization	$16.95
Cosmic Crystals	$19.95
Rebel Defender	$19.95

We also carry the full range of Commodore Cartridges, Disk and Cassette Programs.

AND LOOK AT THESE PRICES FOR JUNE & JULY ONLY

ATARI

Atari 800 48K	1225.00
Atari 400 48K	639.00
Atari 410 Recorder	139.00
Atari 810 Disk Drive	799.00
Atari Joystick	36.00
Quickshot Joystick	29.95
Atari 822 Thermal Printer	439.00
Atari 400 48K plus keyboard	799.00

ATARI SOFTWARE CARTRIDGES

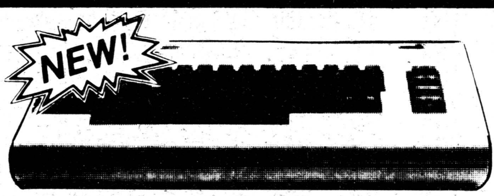
Packman, Star Raiders, Galaxians, Defender, Centipede, Space Invaders, Asteroids, Missile Command, Pilot with Turtle Graphics, Computer Chess, Music Composer, Basketball, Super Breakout, Video Easel.

AND NOW MOON BASE 10

Talk to and hear your Earth base while fighting on Jupiter's moons. A new super fast, high action, adrenoline pumping space game.
Introductory offer $29.95 (Cass)

COME FOR A DEMONSTRATION ON THE NEW COMMODORE 64

commodore



COMMODORE 64

The Commodore 64 is a compact unit that can even fit into a brief-case. Complete with these features:

- Full size typwriter keyboard
- Full ASCII character set
- Upper and lower case
- High resolution colour graphics
- 40-column video display
- 64K RAM under memory (standard)
- CP/M operating system option
- TV Modulator interface
- Game cartridge slot
- Music Synthesizer
- Smart Peripherals

PLEASE CALL US FOR MORE DETAILS

We already have an enormous range of Commodore 64 Software including adventure games, action space games, word processors, educational and business programs.

The Largest Range of Vic 20 Software in Australia

LLAMASOFT UK

ANDES ATTACK

As a star warrior of the Galactic Fleet your mission is to guard LLAMAS who are under attack by the Alien Xygar Life Forms. In emerencies use your precious smart bombs.
Requires VIC-20 with 8k expansion and joystick.
On Cassette

ABDUCTOR

Guard your humanoids from the Abductors flying in crazy loops! Your humanoids look worried. Get blasting!
Written in machine code. For the unexpanded VIC-20 with joystick.
On Cassette

GRIDRUNNER

You are the Gridrunner. Your mission is to destroy the Gridsearch Sqads and the pods. But beware of the deadly X-Y zappers.
Written in machine code. Hires colour graphics for the unexpanded VIC-20 and joystick.
On Cassette
Also available for the Commodore-64.
(Please specify version)

TRAXX

Your mission is to capture squares for points. When pursuers flash eat them for bonus. Capture the 4 corner squares to make pursuers flash.
Written in machine code. Requires VIC-20 with 8k expansion and joystick.
On Cassette

CREATIVE SOFTWARE

Choplifter	$54.50	(Cart)
Apple Panic	$54.50	(Cart)
Serpentine	$54.50	(Cart)
Trashman	$54.50	(Cart)
Astroblitz	$54.50	(Cart)
Video Mania	$54.50	(Cart)

EDUCATIONAL

Over 100 Vic 20 Educational programs now available

ADVENTURE FANS

Come and get some free advice on our Scott Adams, Epyx and other great adventures.

ALSO SUPPLIERS FOR:

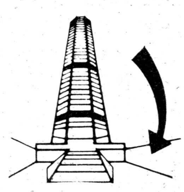
U.M.I.
Contronics
Epyx
Saturn Soft
C.W. Electronics
Abacus
Cartersoft
Vic Soft
and many more

VIC-20 at

the Computer Spot

Sydney's own VIC specialist

- Friendly Service
- Lowest Prices
- Expert Advice
- Convenient



Shop C4, M.L.C. Centre
Martin Place
Sydney. 235-2971

Now, That's Really Cosmic!

Cosmic Software is a relative newcomer to the scene of software distribution for the TRS-80 Models I and III and the System 80. Here Eduardas M Grigonis examines some of the games packages from the fledgling company.

COSMIC SOFTWARE specialises in Australian-written games software, but the New South Wales-based company is also expanding into the importation of quality games from the United States.

Each of the five games I am reviewing was originally obtained on tape, and all loaded easily. However, Cosmic is currently in the process of preparing disk versions of all its Australian software.

I used the LDOS CMDFILE utility to transfer the programs to disk, with an appropriate offset. DOSPLUS users can use the TAPE utility and NEWDOS users can use LMOFFSET.

Cosmic Software has available a Load Module Offset program for TRSDOS users. This costs about $5 when purchased with any other program, or $19.95 separately.

Unlike many companies, Cosmic Software has a commendable upgrade policy for games software. Anyone who wants to upgrade to a later version of a game – for example, Version 2 of Rally Racer – can simply return his original tape and $2 to the company.

Many of the games offered are also being upgraded to provide joystick compatibility. (Unfortunately, Defence Penetrator – reviewed here – isn't one of them.)

At the time of writing this review, the prices for each of Cosmic Software's Australian programs was $19.95. At the risk of upsetting some distributors I should point out that people buying direct from the company can expect to receive a voucher entitling them to a $5 credit against their next purchase from the company.

Cosmic Software has established itself in a short time as a creditable source of interesting games programs for the TRS-80 Models I and III and the System 80.

If you have a penchant for games, then I recommend you have a look at what Cosmic Software has to offer.

Star Cresta

When I first decided to write this review, some months ago, I intended to begin with something like, "If this had been the only program I bought at the time, then I would probably have never bought another program from Cosmic Software. Fortunately, at the same time, I also bought Defence Penetrator . . . "

I have since changed my mind. This is not to say that I don't still have some doubts about the program. However, in the interests of giving a fair review, I resurrected it from the months of disuse that resulted from my initial disappointment. Lo and behold, after spending a fair amount of time with Star Cresta, I began to appreciate its finer points.

Based on that arcade favourite, Firebird Phoenix (or was it Moon Cresta?), Star Cresta requires that you succeed in destroying three successive frames of aliens before getting a chance at the massive mothership.

In each of the first two frames, you're up against a dozen small 'birds' which first appear in a standard formation – depending on the frame – and then proceed to fly all over the screen whilst you try to destroy them. These small 'birds' are worth 80 points each.

The third frame presents you with 12 large birds which are each worth 150 points. To destroy them, you have to hit them right in the centre. Unlike the original arcade game, you don't get points for shooting their wings off.

If you manage to make it through to the fourth frame, you find the mothership. This is worth 2000 points, but it will take many hits before it is destroyed. The first problem is that you have to shoot through the bottom shield of the ship. Once this has been achieved, you have to shoot through a revolving panel. If you can then manage to shoot through the holes you have made in the revolving panel, you eventually get through to the occupant of the mothership, which results in its destruction. You then return to the first frame and try to rack up even more points.

The controls for this game are basic. Movement is controlled by the left and right arrow keys, and firing is achieved by pressing the space bar. If the space bar is kept down, firing is continuous,

though only one of your shots may be on the screen at any time.

You also have the option of pressing the @ key or the ? key to apply a force-shield to stop the aliens shots from hitting you. That's right! The aliens are firing back at you, and they have no qualms about having a multitude of shots on the screen at the same time.

An extra ship is awarded at each multiple of 10,000 points.

There are a couple of problems with the game. One 'feature' is a 'realistic scrolling starscape', but I have found this nothing but a nuisance. As a result of this feature, it is sometimes impossible to know when you are being shot at! There is extreme frustration when, suddenly and for no apparent reason, you're blasted from the skies. This scrolling starscape would be most impressive when going between frames but on a black-and-white screen it has no value during actual playing of the game.

In a game of this type, it is virtually standard practice that, after being hit, you return to where you left off. In Star Cresta, however, you return to the beginning of whichever frame you got wiped out in. This was one of my major sources of frustration when I first tried the game.

Is this program worth buying? If you like arcade games, yes, go out and buy it. Be prepared for some initial frustration but you'll eventually find yourself engrossed.

My highest score so far is 116,250, which was achieved well into the 19th round.

Defence Penetrator

Defence Penetrator is Cosmic Software's version of the Scramble arcade game. Your mission is to navigate an alien landscape and destroy the enemy's command headquarters.

The first item you are presented with is the title page. This provides details of possible scores – fuel tanks are worth 10 points and provide extra fuel, missiles are worth 30 points, laser cannons gain 50 points and the command base is worth 2000 points. Reactors can be worth any amount, but I usually get about 300 or 400 points. Some reactors are hidden below the surface, so you can sometimes gain points for bombing seemingly empty terrain.

The program doesn't contain any built-in instructions. Direction is controlled by the four arrow keys. Diagonal movement is possible – however, the left and right arrows really only slow down or speed up your craft, rather than allow changes in direction. The trend is for flight to the right.

The space bar fires torpedoes to the right of your craft; only three torpedoes

can be on the screen at the one time. Bombs are dropped by pressing either of the shift keys; only two bombs can be on the screen at the one time.

The game may be aborted at any time by pressing the 8 and 9 keys.

The mountains are the first part of terrain that you have to overcome. As you fly over this area, missiles are launched at you. You gain points for destroying the missiles and also for dropping bombs on the fuel tanks and reactors, which progressively appear.

The next part is lower terrain, where you will again find fuel tanks and reactors. The missiles have been replaced by laser cannons, the shots of which you have to avoid. You also find enemy craft coming at you from the right of the screen. These aren't easy to destroy.

The third level replaces the enemy craft with meteors. These are impossible to destroy and must be avoided.

If you make it this far, the next step involves entering an underground silo complex, containing missiles, fuel tanks, reactors and laser cannons.

The final stage is a short, hilly area, containing the same obstacles, which leads to the command base. Your aim needs to be accurate to hit the command base.

There is no provision to store the highest 10 scores – the game simply keeps a record of the highest score obtained so far. I would prefer to see provision for more high scores. Apart from that aspect I have no real complaints about 'Defence Penetrator'.

Stellar Warp

The third program reviewed is Stellar Warp. Containing similarities to the Moon Cresta arcade favourite (or was it Firebird Phoenix . . .), Stellar Warp is the one program of the five reviewed that I feel would stand the most chance of selling reasonably well in the United States.

This is the only program of the five with voice sound effects in addition to normal sound. The vocabulary is limited (five words only) but it is good to see an Australian programmer trying such techniques.

The game does not include instructions or scoring details in the actual program. Scores are gained for Razor Blades (30 points), Pods (50 points), Spankers (80 points) and Solaroids (100 points).

These craft are attacking you from the top of the screen and you need to move under them using the left and right arrow keys before firing with the space-bar. Firing is continuous if the space-bar is held down although you are restricted to only having three shots on the screen at the same time. The aliens are also shooting at you and have the ability to fire diagonally.

You will also occasionally find the Obliterator moving down the screen. This cannot be destroyed by your shots but it will eventually self-destruct and throw out gamma particles which will either reduce your shield power or destroy your ship.

The status of your shield is indicated at the top centre of the screen. A whole line indicates that the shield is in reasonable condition and you can therefore sustain hits without being destroyed. A broken line indicates that a hit could be fatal. If the line is broken it can be restored if you manage to go for a while without being hit.

A good option when in difficulties is to hit the enter key. This activates Stellar Warp which moves you up the screen and enables you to ram the alien craft. As there is only one warp per ship, this option is best left as a last resort.

The game may be aborted at any time by pressing the '8' and '9' keys. Extra ships are awarded at every 10,000 points.

If you are determined to buy a program this week then I strongly recommend Stellar Warp.

Provision is included to show the current Top Ten scores and the initials of the person responsible.

Rally Racer

The reason I originally decided to write this article is that I was asked to review version 2 of Cosmic Software's Rally Racer. (I guess being in charge of the Adelaide Micro-User Group's Gaming Sub-group has its advantages!) I originally purchased version 1 of Rally Racer some time ago and I was interested to see how the new version rated.

When you first run Rally Racer you will notice a source of dissatisfaction. The program comes with a high score built in. Although this is now only 10,000 points (as opposed to the high score of 20,480 in version 1) I consider that as

no provision is made in the game for other than the highest score and the most recent score, the inclusion of a built in score that will take some time to exceed is unwise and a source of frustration to the user.

You will also find that, unlike version 1 which has only one speed, version 2 gives you the option of playing in Trainee Mode (press T) or Road Warrior mode (press R). As the standard speed of the program is fast, the provision of a slower speed to enable familiarisation with the program is welcome.

The object of Rally Racer is to steer your car through a maze and attempt to hit flags to score points. At the same time you have to avoid Morgan the Mad Motorist and Crazy Harry and his hoodlums. Morgan can go to any part of the maze (he can go through walls) whereas the others will follow the most direct route leading them to you.

Occasionally you will find an oil slick in the maze. If you run into the slick this will cause you to crash. If you are being chased by the other cars you have the option of creating a smoke screen by pressing the space-bar. This will slow down your pursuers but will also use up fuel.

On balance, I would have to recommend Rally Racer to any of you who are looking for something different from your normal run-of-the-mill arcade games.

Morgoth – The Arcade Adventure

Of the five games reviewed, Morgoth is the one which I enjoy the most. If you can rationalise an 'arcade adventure', then Morgoth is for you.

The object of the game is to kill monsters, collect treasures, avoid Morgoth and complete as many sets as possible. Movement is made by pressing any of the four arrow keys. Firing of arrows is achieved by pressing the spacebar and either the left or right arrow. You can only shoot left or right although all you machine-language buffs out there will be interested to know that the code to enable firing up or down is still in the program and has simply been disabled. It was found that leaving this facility in detracted from the playability of the game.

The game is set in the realm of Thrandabar. You begin in The Caverns Of Moria. You will find treasures scattered about, indicated by a hash symbol. Occasionally you will also find bonus treasures indicated by a question mark. The values of the treasures are gained simply by moving over them. You will also find fireballs, which are indestructible and to be avoided, and Morgoth.

If you survive the Caverns you should try to escape into Smeagor's Keep. You will find more treasures, accompanied by Morgoth (he gets around) and monsters. Eventually you will find your way out of the Keep into the Forest Of Mirkwood. This contains Morgoth, more monsters and treasures and trees. Should you be unlucky enough to become entangled in the trees, escape is impossible.

The only creatures you can kill are the monsters. Anything else should be avoided. Morgoth will only succumb to the most powerful of magic weapons, which I take to mean the average mortal has no hope. Trees are also deadly, not to mention mobile.

Morgoth comes with provision to store the current Top Ten scores. Although there are already ten random high scores built into the program, none of them is insurmountable. Some thought has gone into this element of the game as you will find yourself trying to better the scores of such luminaries as Conan, Hammerhead and Firefist.

Morgoth is a successful attempt to provide something different in the way of arcade games for the TRS-80 and is highly recommended. Although technically not up to the standard of Stellar Warp, it is more enjoyable. □

your computer text file

From Page 61

especially as all BASICs that have PEEK and POKE will generate an error message if their range is exceeded.

ERIC B LINDSAY
Faulconbridge, NSW

And Another One

MY FATHER purchased a Dick Smith Super-80 computer in October 1981. We now have a machine with 64 kilobytes of memory, of which 12 kilobytes is ROM containing Monitor and BASIC. Before the ROM BASIC was purchased, we loaded the tape BASIC at either 300, 600, or 1200 baud. The speed is software selectable, by changing the addresses containing the speed indicators.

The Super-80's BASIC has some excellent commands, especially for string handling (some are easier than MicroSoft).

The Super-80 Monitor also has a CKSM error report for tape loads. Other commands include L (tape load), S (tape save) and I and O (Z80 in/out port read/write).

I have operated MicroBee, Sinclair ZX-81, Apple II, Tandy TRS80 and Canon BX-1 computers, yet I still find the Super-80 a pleasure to use because of its faultless keyboard, no crashes, and outstanding error-free tape-load and save.

DAVID THURSBY
North Rocks, NSW

Join The Peach Club

ALLOW ME to inform your readers of the existence of a club for the Hitachi Peach microcomputer. The club is known as the New South Wales Peach User Group and membership is open to all Peach owners. Outer New South Wales memberships are also taken; all memberships are for a six-monthly renewable period.

A fee of $10 is charged to every member. The fee entitles members to newsletters, access to the club software and technical library, and technical advice.

Meetings are held on the first Saturday of each month from 2pm at Cybernetics Research, 120-122 Lawson Street, Redfern, 2016 (phone 02 698-8286 Sydney).

DANIEL SOUSSI
Redfern, NSW

No One Goofed!

I WAS disappointed to see Bill Visser's letter, headed "You Goofed!", in the January/February edition of *Your Computer*.

I take pains to try out my material on computer club members, to ensure that it is suitable for beginners to microcomputing. This was done with the material to which Mr Visser objects, my October 1982 article on MicroBee BASIC.

When I re-read the article, I found I could not really understand most of Mr Visser's objections. The problem, I believe, is that he was expecting a somewhat deeper treatment of the matter, at a point where an explanation of binary and hexadecimal conversions would have been required to carry the topic forward.

My own experience with beginners is that this is counter-productive. I simply did not feel it appropriate to introduce too much detail in a simple article.

Mr Visser states that it is mandatory to store data in reversed format, and objects to my use of the word "easier". It is true that if you use the Z80 double-registers, you will want the data in reverse format. However, any sufficiently perverse programmer can write an assembler program that doesn't use data in this format. It is not mandatory, merely easier, to write low byte rather than high byte.

Mr Visser is correct to state that PEEK and POKE are limited to 255 down to 0 because of the eight-bit memory space. However, to explain that again involves covering a bit about binary. I find it better to blame BASIC, especially as all BASICs that have PEEK and POKE will generate an error message if their range is exceeded.

Well, No One's Perfect

MY PROGRAM in the December 1982 article "MicroBee With Character" has been entered into our computer as printed in the magazine, and some errors have been discovered.

Lines 340 and 340 should obviously have different numbers.

Line 1150 should read *IF D(*and so on, and not *IS D(*.

Lines 570, 580, 590 and 600: Where the square-bracket construction is used to pass parameters to the sub-routines, the numbers should have a comma between them.

Line 840 has two more spaces than needed in the ;literal.

One line seems to have been omitted – the program will work without it, but the column and row numbers won't be displayed at the top of the screen, as intended:

975 CURS 20: PRINT " c "; [12 C];" r "; [13 R];

With this changes made, the program operates correctly.

HARRY PURVIS
Regent's Park, NSW

The Wrong Stuff

IT WOULD BE a fair assumption to say that the majority of your readers purchase the magazine because of the program data published therein.

However, such an advantage is lost if the data is illegible, because the typeface is too small and the characters over-inked.

Programs can take much time to enter . . when I'm forced to assume that illegible data is such and such, and my assumption proves to be wrong, then I would wish you to be close by.

ALAN N STEPHENS
Belmont, Vic

RANKED IN THE US TOP 30!
Each product marked ☆ is ranked in the top 30 list of the best IBM PC software rated monthly by the authoritative US magazine "Softalk"

New Releases from Sourceware's Software Supermarket

Sourceware is out to prove it has Australia's finest range of exclusive software for the IBM PC user. Here's just some of the top-performing unique software just released by Sourceware. Get the full details from your IBM PC dealer, or ring Sourceware for information or your nearest stockist.

IRMA Board and IRMALINE ®

Terminal PC board and terminal emulation for the IBM 3270. IRMA is a Decision Support Interface, a super sophisticated circuit board that fits in any slot in your IBM Personal Computer. It permits you to access and store-mainframe data in the privacy of your office, whenever it's convenient for you. IRMA will assist your IBM PC by providing better data, more quickly and efficiently. It even comes with a documentation pack that fits neatly into your PC manual.

☆ LOTUS 1-2-3 ®

Rated No. 1 on "Softalk's" monthly rating of the best IBM PC software. With 1-2-3, you can be using your IBM PC in a surprisingly short time. Even if you've never been near one before. 1-2-3 instructs you right on the computer's screen in a friendly, helpful way, so you learn as you go along. Everything is in English, not code, so there's no new language to learn. There's even a special HELP key you can press to put special instructions on the screen if you can't remember what to do next. But for all its comforting ease of use, 1-2-3 is one of the most powerful programs available for the personal computer. It combines spreadsheet, information management and graphics all in one.

☆ CROSSTALK ®

Crosstalk is a data communications program that opens lines of communication between your microcomputer and virtually any other computer. That includes other microcomputers, remote mainframe computers, and subscription information services such as The Source and Compuserve. Crosstalk is suitable for almost any CP/M, CP/M-86, MS-DOS or PC-DOS based microcomputer, Crosstalk is available for most popular 8 and 16 bit computer systems.

SWEET-P ® Colour graphics plotter

Sweet-P is a high speed, high quality graphics department right at your fingertips. Sweet-P draws superior hard copy graphics with super-fine plotting resolution and line quality. Virtually every graphic format you will ever need is provided – colourful pie charts,bar graphs,straight and curved line graphs, and illustrations and alphanumeric labelling. What's more, with commercial/graphics software, mountains of statistics can be reduced to clear, concise business graphs. Once stored in memory, any graph can be easily edited and redrawn.

DEALER ENQUIRIES WELCOME

Here's more new releases from Sourceware (see your IBM PC dealer for more details, or ring Sourceware direct):

SO2.FPC

☆**EASYWRITER 11**	Second Generation Word Processing System
EASYFILER	Data Management System
EASYPLANNER	Electronic spread sheet
EASYSPELLER	Spelling dictionary
☆**PC TUTOR**	Complete PC user training system
PC PAL	PC education
☆**FRIENDLYWARE**	Intro set for new PC users
PC ARCADE	Arcade games
PC1051	IBM SYS 34/38 – PC interface
PC1076	IBM SNA/SDLC – PC interface
SIDEWAYS	Vertical print utility
☆**TIM111**	Database manager
FAST GRAPHS	Graphics package
PC DOCUMATE	Keyboard templates, DOS 1.1 & 2.0, Basic, Wordstar, Visicalc, Multiplan, Base 11, Easywriter II and Lotus 1-2-3

SOURCEWARE

4/73 Albert Avenue, Chatswood, NSW 2067. (02) 411 5711

The Source of Software

If you need software, simply come to the source

SOME COMPUTERS ARE BETTER THAN OTHERS

COMPUTER COMPARISON CHART

	SPECTRAVIDEO SV-318	APPLE II PLUS	ATARI 800	COMMODORE 64	COMMODORE VIC 20	TANDY TRS-80
BASE PRICE	**$499**	**$2100**	**$1100**	**$699**	**$299**	**$549**
COMPUTING POWER FEATURES						
BUILT-IN ROM	32K	12K	10K	20K	20K	8K
EXPANDABLE TO	96K	N/A	42K	N/A	N/A	14K
BUILT-IN EXTENDED MICROSOFT BASIC	YES	YES	ADDITIONAL COST	NO	NO	NO
BUILT-IN RAM	32K*	48K	16K	64K	5K	4K
EXPANDABLE TO	144K**	64K	48K	N/A	32K	32K
KEYBOARD FEATURES						
NUMBER OF KEYS	71	51	61	66	66	53
USER DEFINE FUNCTIONS	10	N/A	4	8	8	N/A
SPECIAL WORD PROCESSING	YES	NO	NO	NO	NO	NO
GENERATED GRAPHICS (FROM KEYBOARD)	YES	NO	YES	YES	YES	YES
UPPER/LOWER CASE	YES	UPPER ONLY	YES	YES	YES	UPPER ONLY
GAME/AUDIO FEATURES						
SEPARATE CARTRIDGE SLOTS	YES	NO	YES	NO	NO	NO
BUILT-IN JOYSTICK	YES	NO	NO	NO	YES	NO
COLORS	16	15	128	16	16	8
RESOLUTION (PIXELS)	256 x 192	280 x 160	320 x 192	320 x 200	196 x 184	192 x 256
SPRITES	32	N/A	4	8	8	N/A
SOUND CHANNELS	3	1	4	3	3	1
OCTAVES PER CHANNEL	8	4	4	9	9	3
A.D.S.R. ENVELOPE	YES	NO	NO	YES	YES	NO
PERIPHERAL SPECIFICATIONS						
CASSETTE	2 CHANNEL	1 CHANNEL	2 CHANNEL	1 CHANNEL	1 CHANNEL	1 CHANNEL
AUDIO IO	YES	NO	YES	NO	NO	NO
BUILT-IN MIC	YES	NO	NO	NO	NO	NO
DISK DRIVE CAPACITY	256K	143K	96K	170K	190K	156K
(LOW PROFILE)	YES	NO	NO	NO	NO	NO
CP/M COMPATIBILITY (80 column programs)						
CP/M* 2.2	YES	NO***	NO	NO****	NO	NO
CP/M* 3.0	YES	NO	NO	NO	NO	NO

* 16K user addressable plus 16K graphic support
** 128K user addressable plus 16K graphic support
*** Apple II can accept modified 40 or 80 column CP/M
**** Commodore 64 accepts 40 column CP/M

Microsoft is a registered trademark of Microsoft Corporation
CP/M is a trademark of Digital Research, Inc.

OURS IS MUCH BETTER

When you start comparing Spectravideo's SV-318 to other personal computers, you'll find there really is no comparison. The SV-318 is the only logical choice, because it does more than some computers costing 4 times as much. And its abilities simply embarrass other computers in this price range.

The SV-318 isn't just more capable. It's much more capable. No other computer at even twice the price comes near its 32K ROM expandable to 96K. Or to its 32K RAM expandable to 144K. And no other computer has a built-in joystick/cursor control—an immeasurably useful feature when it comes to playing your favorite video game. Further, the SV-318 has, as its resident "language" Extended Microsoft Basic, the industry standard. It even has built-in CP/M (standard 80-column program), so you can immediately utilize over 10,000 existing software programs.

The SV-318 isn't just more expandable. It's much more expandable. Unlike many other so-called computer systems, all our important peripherals are available at once. That means you can get almost full usage out of your SV-318 from the day you buy it. With the Super Expander, Data Cassette, Floppy Disk Drive, Dot Matrix Printer, Graphic Tablet and SV-800 Series Expansion Cartridges, there's almost no end to the work you can do. Or to the fun you can have. The SV-318 is well designed to interface with new options as they become available, too. All this adds up to a computer you'll grow into, not out of.

The SV-318 is not only eminently affordable, it's the first real bargain of the computer age! Besides business application, home budgeting, word processing, programming and self-teaching, the SV-318 is the best entertainment value in town. Not only can you use it with your TV or color monitor to play hundreds of different video games,

FOR UNDER $500

with the optional SV-105 Graphic Tablet you can draw pictures, graphs, charts and other visual images on your TV screen. Considering what you get for what little you pay, the SV-318 is once again the only logical choice.

Whether you're investing in your first computer, or are already well versed in today's most important machine, you'll find that the SV-318 is the only logical choice for you.

SPECTRAVIDEO™

SV-318

PERSONAL COMPUTER

VIDEOACTIV ELECTRONICS
70 St. Kilda Road, St. Kilda, Vic. 3182. Ph: (03) 537-2000
P.O. Box 474, Edgecliffe, N.S.W. 2027. Ph: (02) 328-1190

Getting dBest From dBase Part II

Les Bell continues his examination of dBase II with a look at the full screen editing functions . . .

LAST MONTH WE looked at the creation of a database file, and left you looking at the file using various dBase II commands. This month, we'll continue by looking at the various full-screen commands in depth, together with the editing functions which dBase provides.

Incidentally, one of the things I continually moan about are people who write BASIC programs and then don't say for which version of BASIC on what machine they are writing. Last month I broke my own rule by not telling you what version of dBase I was using. All examples in this series will work on version 2.3 and later. I'm currently using version 2.31, with 2.4 due to arrive shortly, and I'll apprise you of any differences between versions where necessary.

Last month, we created a database of book titles and stocks. Right now, I'd like you to fire up dBase and USE that file, BKINV. If it's on the disk in drive B:, then type USE B:BKINV.

If you now type APPEND, dBase will position itself to the end of the file and will draw up a blank form on the screen, ready to be filled in. Go ahead now and enter the title of a book and fill in the rest of the form. As you go, try experimenting with the following keys:

Control-D: Moves the cursor one character to the right. If it's at the end of a field, it will move on to the next field. The right arrow key on your terminal may have the same effect.

Control-S: Moves the cursor one character to the left. If it's at the beginning of a field, it will skip back to the beginning of the previous field. The left arrow key on your terminal may have the same effect.

Control-E: Moves the cursor up to the beginning of the previous field. The up arrow key on your terminal may have the same effect.

Control-X: Moves the cursor down to the beginning of the next field. The down arrow key on your terminal may have the same effect.

DEL: Deletes the character behind (to the left of) the cursor.

Control-G: Deletes the character under the cursor.
Control-V: Toggles insert mode on and off. In insert mode, any characters typed into a field will 'bump along' those to the right; with insert off, new characters will simply overwrite old ones.
Control-P: Toggles printer output, in the same way as it does on the CP/M command line.
Control-Q: Abandons the current operation and returns to dBase command mode with no changes made to the current record.
Control-R: Writes the current record to disk and moves onto the next one (Note that this has a different meaning in edit mode).
Control-U: Marks the current record for deletion – pointless in append mode, but it does work!

Notice also that an empty entry on the first line will return you to command mode; this is the standard way of telling dBase that you want to finish appending.

Using these commands, you have considerable flexibility in inputting data; no matter how bad a typist you are, you can generally clean up your input as you go along, to provide neat data.

However, there are a few things you can do to make life easier for yourself. For example, if your application requires that you enter several invoices at a time, all to the same company, you can organise dBase to carry over the data from one record to the next as you append, and you need only re-type the altered data.

To do this, you use the SET command. dBase has many parameters, or default modes of behaviour, which are alterable using the SET command. To get dBase to carry fields across during APPENDs, use the SET CARRY ON command. You might like to try that now.

There are several other defaults you can use, and we'll introduce these as we go. But here are some of the more generally useful ones.

For example, you may have noticed that if you enter data right up to the end of a field, dBase will move on to the next field automatically, and beep at you to warn you. Perhaps the beeping or ringing of the bell might annoy you or people working nearby.

Here's your big chance, Mr. Whelan! The (dare I say it) SET BELL OFF command will silence the bell. If you don't want dBase to move on to the next field automatically, as you tend to just keep typing and put the characters into the wrong field, you can tell dBase to SET CONFIRM ON. Now it will not advance to the next field until a control key (one of the editing keys or RETURN) is typed.

dBase naturally wants to display its prompts in half intensity, and will send attribute characters to the screen to do this. However, your screen may not be happy about this, and may display Greek characters or whatever; this certainly happens on the Kaypro. To avoid the problem, use SET INTENSITY OFF, and dBase will refrain from sending those characters. It will still send the occasional null, but the screen will be much cleaner.

By the way, if your terminal does not support cursor addressing and clear screen commands, you can still make use of dBase, in a limited way. Type SET SCREEN OFF, and dBase will no longer use the full screen operations. In this mode, append will scroll fields up from the bottom of the screen, and EDIT becomes a bit trickier to use, as we'll see later.

If you want a copy of all your output sent to the printer, then type SET PRINT ON. To turn the printer off again, type SET PRINT OFF. If you have no printer attached to the system currently, you can still create a transaction log by echoing output to a disk file.

If you type SET ALTERNATE TO <filename>, then later type SET ALTERNATE ON, dBase will create a disk file named <filename>.TXT, and will send all output there until a SET ALTERNATE OFF command is issued. This is particularly useful to authors of articles about dBase!

If, for some reason, the colons (:) which mark fields as you append annoy you, you can get rid of them with the SET COLON OFF command. These commands can be reversed (ON/OFF) and default to practical values, which are listed in the manual.

One of the things on my dBase wish list is the ability to preset these at installation time so that dBase powers up the way you want, but as we'll see later, it's easy enough to create a command file to set them as required.

Database files are saved with up to eight characters of filename followed by the extension .DBF. You can specify the drive they will be stored on, if it's not A, by prefixing the name with a drive code like B: or C:. However, rather than keep doing this all the time, you can set the default drive that dBase will search for files by using the command SET DEFAULT TO B (or whatever). From that command on, dBase will expect all files to be on drive B, unless otherwise specified.

Editing Data

That covers appending to a database. You'll have noticed by now that dBase offers you the chance to immediately append to a database after you've created it, and after you have appended it will return to command mode. After this, or when you want to work in any way with a database, you must select that database with the USE command.

So the command USE B:BKINV will open the file B:BKINV.DBF and make it ready for use. dBase will read the first record, which contains information about field names, so that it can make sense of your commands.

If you decide to edit a record – say, to update an address – just type EDIT <recordnumber>. dBase will now display the record in the same format as APPEND, and you can move the cursor around and make changes as with APPEND. Virtually all the editing controls listed above for APPEND are still available, with the following changes and additions:

Control-R: In edit mode, this writes the current record to disk and backs up to the previous one.
Control-C: Writes the current record to disk and moves on to the next.
Control-W: Writes the current record to disk and returns the user to command mode.

Notice the remarkable resemblance between the dBase II editing controls and those of WordStar; the two are logically similar, and where there are differences, they make sense in the light of the database's record orientation.

Editing with full-screen operations disabled is possible, but not nearly as convenient. To edit a particular field of a record, you must type EDIT <recordnumber>, <fieldname> or EDIT <recordnumber>, <fieldnumber>. dBase will respond by showing the current contents of the field, and asking what you want to change, and what you want to change it to.

It's a tedious process in comparison with full-screen editing, but it's workable. Some day, probably soon, you will find yourself operating your system remotely, using a hand-held terminal with only a few lines of liquid crystal display. Then you'll be glad dBase II has this capability!

What's In There?

You now have the basic ability to create and update a database – any database. The trick now is to get the information you have put in there back out again. There are several ways to do this, but the most useful is DISPLAY.

DISPLAY will display the contents of the current record, all records, all records that meet a selection criterion, and so on. Its basic format is this:

DISPLAY [<scope>] [FOR <exp>] [<exp list>] [OFF]

The square brackets ([]) surround optional parameters, while the angle brackets (<>), as always, indicate non-literal items. Let's look at these in depth.

Many dBase commands will accept a scope. As you'd expect, the scope indicates the range of records over which the command will work. At any moment, dBase II is positioned to a particular record, and the DISPLAY command will either operate on that record (the default if no scope is specified), the next *n* records, all the records in the database, or some other specified record. The scopes are as follows:

ALL: All the records in the file.

NEXT n: The next *n* records in the file, including the current record; *n* must be a literal constant, not a variable.

RECORD n: only record *n*; and again, *n* must be a literal constant for this to work.

In the absence of one of these scope modifiers, DISPLAY defaults to the current record. The FOR ‹expression› clause allows the user to select only those records which meet specific criteria. An expression, in this case, must have a logical value; in other words, it must be true or false. Here are some examples:

```
BUY:PRICE < 10.00
AUTHOR < 'M'    STOCK < REORDER
PUBLISHER = 'HRW'
```

Unless otherwise specified, the presence of a FOR clause will cause a dBase command to default to a scope of ALL.

The expression list after the FOR clause in the DISPLAY command is generally a list of field names to be displayed. If this is not supplied, then dBase defaults to displaying all fields in the selected records. Finally, dBase will start each line with the record number, unless the OFF parameter is supplied.

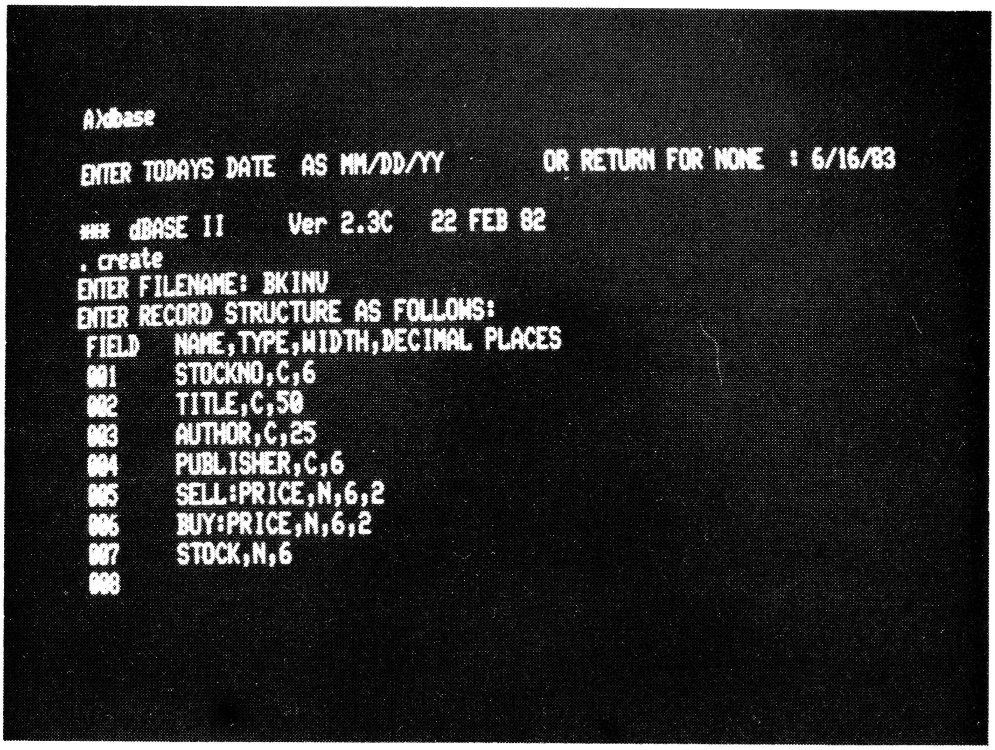
So, here are some valid DISPLAY statements for our sample database:

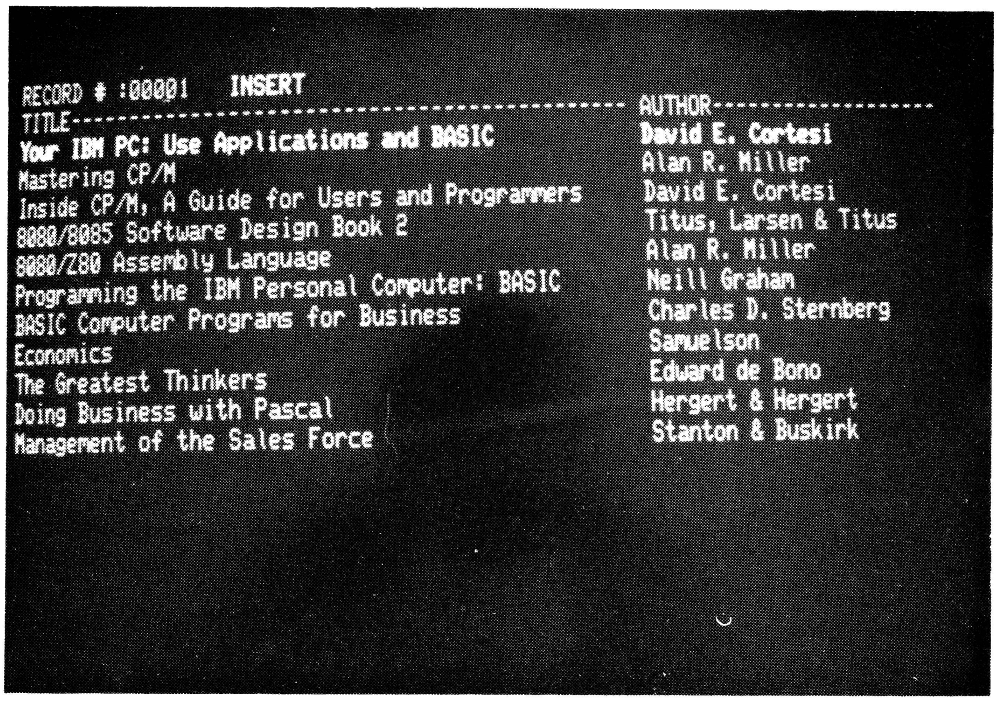
```
DISPLAY FOR STOCK < REORDER STOCKNO
DISPLAY ALL TITLE, AUTHOR OFF
DISPLAY FOR PUBLISHER = 'HRW' AUTHOR, STOCKNO OFF
```

dBase is reasonably forgiving in its syntax requirements, too: the commands listed above can be expressed in several different ways, yet would work the same. For example, you get the same result using:

```
DISPLAY STOCKNO FOR STOCK<REORDER
DISPLAY OFF ALL TITLE,AUTHOR
DISPLAY OFF AUTHOR,STOCKNO FOR PUBLISHER='HRW'
```

We'll continue next month with simple reporting. □





Logic For Literati Part IV

In the last article of this series, we looked at the AND, NAND, OR and NOR gates; this month, it's time to start combining them to do something useful.

your computer tutorial

THE BASIC LOGIC gates, by themselves, are not terribly useful; however, they are the building blocks from which everything else, including your own computer, is built. They can be combined in various ways to perform an infinite number of tasks.

For example, one of the fundamental operations all computers must be able to perform is addition. Can we design a circuit which will perform addition? Let's first of all write down the truth table for the addition of two numbers, A and B. The result of the addition will be the sum, S – and don't forget the carry, C.

Now take a look at the S column and see how it compares with the logic functions we have seen before. It doesn't match any of the AND, NAND, OR or NOR patterns, but it certainly does match XOR, the exclusive OR. So, to get the sum of two single-bit inputs, we use the XOR gate, the 7486.

Now take a look at the carry out column; yes, it's the AND function. So, to make an adder circuit, we simply use an exclusive OR gate and an AND gate (see figure 2). Note, however, that there is a fatal flaw with this circuit: it produces a carry out OK, but there is no provision for accepting a carry in. For this reason, this circuit is called a half-adder; at least it points us in the direction of a complete solution.

However, we quite often find that NAND and NOR gates are cheaper than AND and OR, and prefer to design using them wherever possible. The XOR function, though, is so useful it's sometimes worth the extra money. So figure 3 shows another way of producing a half-adder, this time using two NAND gates in place of a single AND. One input of the second NAND is pulled high, and so it will work as an inverter. A spare gate from a 7404 would do just as well.

Suppose, though, that the 7486 is too expensive for the application – or that you have three spare gates left from a 7402 and a couple of inverters left over in a 7404. Can you build a half-adder from this load of left-overs? Yes, indeed you can; the circuit shown in figure 4 will do fine.

Just how this was worked out is not obvious. It leads us to a discussion of negative logic and De Morgan's rules.

So far we have viewed logic in a logical fashion: we've regarded a 1 as being true and a zero as false. What happens if we reverse the meanings of these symbols, so that 5 V is no longer true, but false, and 0 V is true? Before we go on, you might like to think for a while about the truth table and physical behaviour of the 7408 AND gate if we use this negative logic.

Go back to the truth table (figure 1), substitute ones for zeros, and see what kind of truth table you end up with.

In fact, what happens is that the AND gate now changes function, and becomes an OR gate! Likewise, the OR gate reverses function and becomes an AND gate, and NAND, NOR, XOR and XNOR also do the same thing. We can illustrate this in a table:

Positive Logic	Negative Logic
AND	OR
NAND	NOR
OR	AND
NOR	NAND
EXCLUSIVE OR	EXCLUSIVE NOR
EXCLUSIVE NOR	EXCLUSIVE OR

Table 1.

What designers generally do – and it's a useful trick to cultivate – is to switch from positive to negative logic at each stage through a circuit. As mentioned above, NAND and NOR are cheaper

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 1. Truth table for the addition of two numbers.

and more useful than AND and OR, so the designer might visualise the NAND or NOR gates in the first stage of a circuit as being AND or OR gates that happen to produce negative logic outputs. The next stage will use negative logic, and so NORs and NANDs will swap function.

This is illustrated in tabular form in Table 2. This also shows the shorthand designers use for writing down logical equations, in the same way as mathematical equations are written. Two letters written together imply an AND operation, in the same way as an implied multiplication. The arithmetic addition symbol is used as shorthand for the OR function, and the bar over the top is used to indicate NOR or inversion.

You will notice that I'm trying hard to avoid the bar over the top of characters in the body of the article because typesetters cannot directly place those bars, and they have to be manually added later by an artist. Inevitably, even with the greatest care and proofreading, one gets added in the wrong place or left out, and the resulting confusion to readers is a major problem. I've even seen one magazine print De Morgan's rules wrongly!

For these reasons, the IEEE (Institution of Electrical and Electronic Engineers) suggests that an asterisk after a signal name (e.g. A*) be used to indicate negation, and I'll use this standard wherever necessary.

Referring to Table 1, notice that to perform AND with an OR gate, you must invert the inputs, OR these signals, and then invert the output. To use a NOR gate to AND signals is even easier – just invert the inputs, and the gate automatically supplies the final inversion for you!

Figure 4 should make a little better sense this time around. The top NOR gate on the input produces the AND of NOT A and NOT B; in other words, it's an AND gate in negative logic terms. The lower NOR gate performs the same function – a negative logic AND – but this time, it's working on inverted inputs anyway, so the result is the AND of A and B, which is what we want for the carry output.

The final NOR gate is a bit more complex in function, but if you look at the equations at the output and how they are manipulated, it should begin to make sense. Remember that the objective of the circuit is to produce an exclusive OR of the two inputs, which is expressed logically as A*B + AB* (NOT A AND B all ORed with A AND NOT B – remember A*B is (NOT A) AND B). Hope that's not too confusing!

How then can we make a full adder? The circuit is shown in figure 5 and Table 3.

Ci	A	B	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 3. Truth table for a full adder circuit.

The full adder circuit is the basic one found inside all microprocessor chips to allow them to add 8-bit or 16-bit words. To work with 8-bit words will require eight full adders all daisy-chained together – the carry in for the least significant adder comes from the carry bit in the flag byte, and the carry out goes back there.

The full adder is not the only way to add numbers, as we shall see when we come to clocked logic. In the meantime, you are probably wondering how we can design a circuit to perform multiplication. There is a circuit called a binary rate multiplier, but it performs another function, outside the scope of this series.

Fortunately, a clue does survive from Biblical times. It is reputed that after the floodwaters had subsided, and Noah had released all the animals, life started to return to normal as the animals reproduced and the world was repopulated. All except for the two snakes.

When Noah heard about this, he wasn't at all perturbed. He went and found the snakes and brought them back to the Ark, where he placed them on a rough-hewn wooden table. When challenged as to what he was doing, he replied, "Why, everyone knows that even adders can multiply on a log table!"

Thank you, you've been a wonderful audience. Next month, flip-flops, keyboard debouncing and other topics. □

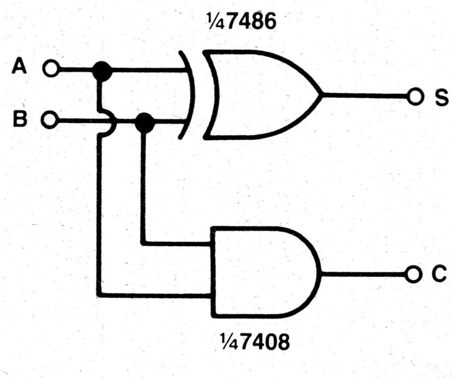


Figure 2. A half-adder circuit.

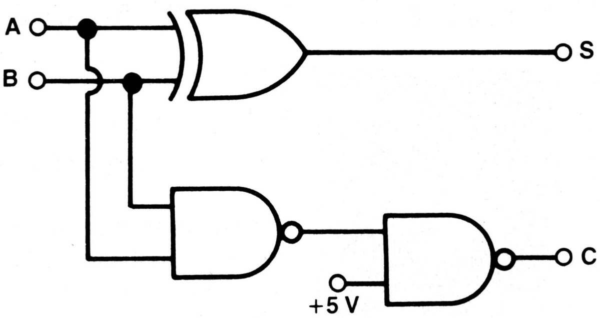


Figure 3. Half-adder using NAND gates.

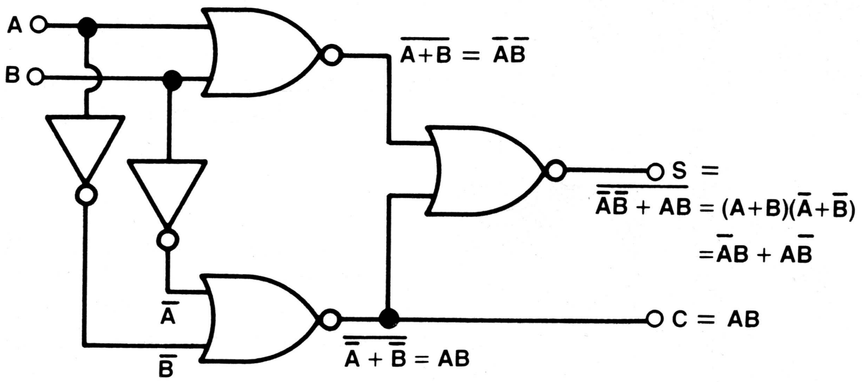


Figure 4. A half-adder made out of NOR gates and a pair of inverters.

$\overline{AB} = \overline{A}+\overline{B}$

$\overline{A+B} = \overline{A}\,\overline{B}$

$\overline{\overline{A}} = A$

$AB = \overline{\overline{A}+\overline{B}}$

$A+B = \overline{\overline{A}\,\overline{B}}$

Table 2. De Morgan's rules.

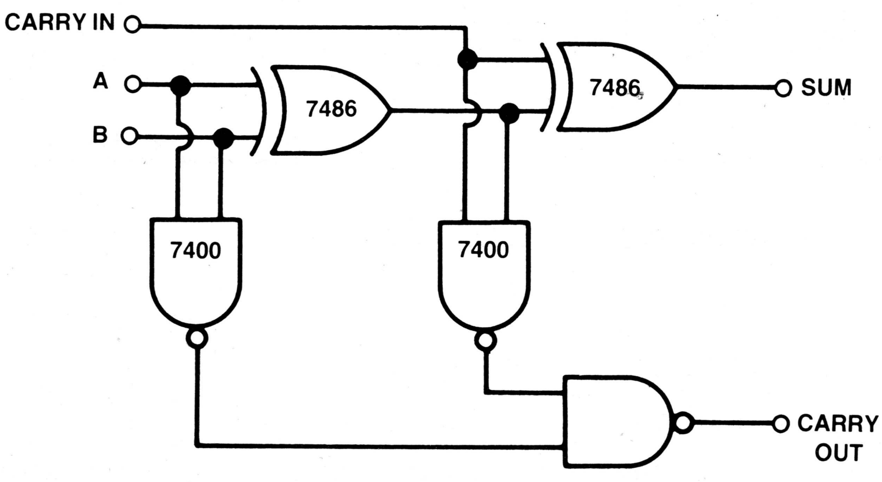


Figure 5. A full adder circuit. The reader may like to try re-designing it using NOR gates.

PAGE 83 is missing

PAGE 84 is missing

By Jeff Richards

THE MANUAL for S-BASIC is adequate as a reference but it's not very useful as a learning guide. To help overcome this deficiency, David Hill, of VLC Australia, has produced a slim volume called *S-BASIC – An Absolute Beginner's Guide.* It shows the beginner how to construct an S-BASIC program – from the text editor through to the compiler.

The book deserves full marks for effort, but it doesn't quite come up to the standard I had hoped for. It attempts to deal with the beginner BASIC programmer as well as the newcomer to S-BASIC. I would have preferred it to have assumed that the reader had already digested one of the *Introduction to BASIC*-type books, and was looking for a hand-holding guide into getting S-BASIC up and running.

The result is that in a scant 56 pages (55 actually – there's no page six), only the simplest features of the compiler and a small subset of the commands can be covered.

The simple commands are covered well, using examples that are complete programs so the beginner can actually watch each new command working. The examples gradually grow into a simple case-book program that is well documented. Because the syntax has been kept as elementary as possible, the sample program makes no use of the advanced control structures available in S-BASIC, which is a pity.

Those struggling with S-BASIC may find the book of some value, but I would suggest you also consider one of the excellent *Introduction to BASIC*-type volumes as a starting point, and then sit down and try to work out the S-BASIC manual. There is also a rumour that the S-BASIC manual is being re-worked, so it may be worth waiting for the revised version.

VLC Australia can be contacted at 172 East Boundary Road, East Bentleigh, 3165, phone (03) 570-5188. The booklet is also available through selected President dealers.

For those using S-BASIC, note that the $LINES compiler directive, which suppressed line-number information in the object code, produces a speed improvement of between three and five times. An excellent review of S-BASIC can be found in *Lifelines*, volume one, numbers 10 and 11.

Imperfect Calc

The LN(x) function in Perfect Calc doesn't work – at least, it doesn't in my version of the program. I don't know what function it implements, but it would appear to make a good random number generator.

President has not yet replied on whether a fix will be available, but Perfect Software is apparently ready to distribute some bug fixes.

American Graffiti

The Kaypro User Group in the United States appears to be alive and well. Its newsletters are chatty and informative but, unfortunately, the editorial staff doesn't identify itself at all, though the standard disclaimer refers to Kay Computers.

Foreign rates are currently $US24 a year (six issues), but there is no indication of whether this is airmail or not (I suspect not). Though billed as a "club" and not just a newsletter, there is no mention of additional facilities that may be of interest to Australian users.

Another American newsletter that may be of interest to Kaypro owners is *Micro-Cornucopia*, the "Single-Board Systems" journal. It started for the Ferguson Big-Board but now includes items for Xerox 820 and Kaypro users. (The editor does all his writing on a Kaypro.) It is larger and a little more technically orientated than the KUG newsletter. The price is US$26 for one year (six issues), by airmail.

The address for KUG is PO Box 100, Malverne, NY, 11565. *Micro-Cornucopia* is PO Box 223, Bend, OR, 97709.

Gamesmanship

MicroSoft BASIC-80 interpreter is now being released as part of the software package for the Kaypro. This represents a significant improvement to the standard software, as BASIC-80 must be the most widely used version of BASIC in small computers.

At this time, it appears the President dealers will not be supplying the new software to existing users, as was done with the Perfect software. However, the chance of existing users being able to talk themselves into special deals should be good.

The BASIC-80 disk comes with three commercially written games and source code for a handful of others.

The three commercial games are:
Ladder, which is like the old standard of barrels rolling down ramps and chutes as you try to reach the treasure at the top;
Catchum, which looks a lot like Pac-man; and
Aliens, which seems very similar to Space Invaders.

All three are supplied in object code only, but are set up for the Kaypro. They use a configuration file so, though the configuration program is not supplied, it isn't difficult to edit the data file.

Aliens is written in C, but I can't identify the language used for the other two. They all operate sufficiently fast to provide a range of skill levels that should satisfy even expert players. The three programs are a good example of just how effectively games can be implemented without fancy graphics.

Version four of BASIC-80 is also supplied. This is useful if version four programs have to be converted to version five. The technique is to load the programs under version four of the interpreter and save them in ASCII format. Then a word-processor can be used to insert all the extra spaces that version five requires. They can then be loaded and saved under version five of the interpreter.

The Word Plus

Another addition to the range of standard software for the Kaypro is The Word Plus by Oasis Systems. It is best described as a package of word-processing utilities, and includes a spelling checker, dictionary maintenance, word-counter, word-usage analysis, word-finder, hyphen help and spelling assistance.

The package is a powerful writing tool, and would be appreciated by anyone doing a lot of writing. For instance, if it finds a misspelled word, it will offer suggestions as to what the word might be.

For the casual word-processor user, the facilities provided are probably more powerful than necessary and, as they take up a whole disk (with 137 kilobytes for the dictionary), you would have to really want to use the procedure to bother with loading it up.

It is quite possible to use only some of the facilities of The Word Plus, which makes it a bit more practical for everyday use. The individual facilities can be called from Perfect Writer (the C command at the main menu) as long as the file to be worked on is nominated.

The Word Plus is a significant addition to the set of standard Kaypro software, but a certain amount of effort is needed to make it usable in a practical environment. □

your MICROBEE computer

By Richard Pakalnis

ONCE AGAIN, I get to eat humble pie (I still haven't met a carbohydrate I didn't like). A letter from the Canberra MicroBee User's Group has set me straight on a few points. Hugh Gibson writes, "Thank you for mentioning our user's group in the June issue. However, I wish to clear up a few errors.

"I am the convenor of the group. I am in no way associated with the MicroBee store in Canberra (at Coolman Court) at a managerial or higher level. I do work there on Saturday mornings – to answer questions and solve problems that any 'Bee owner might have.

"Chris Nicholls is the manager of the store and does not own it. He has no official association with the Canberra MicroBee User's Group.

"The group now has about 80 members and is growing fast. The newsletter is growing, too – the last edition was 30 pages long." Hugh went on to say that membership is open to anybody, not just ACT residents.

I stand corrected and thank Hugh for his letter.

Newsy Newsletters

Hugh Gibson made a good point about membership to user groups that are not in your area. Judging by the quality of the newsletter which he and his Canberra User's Group committee produce, a great amount of information is to be gleaned from its pages, and membership can provide a range of services hitherto unavailable for 'Bee owners.

Colin Tringham and the Sydney MicroBee User's Group (formerly known as the Northside MicroBee User's Group) also deserve praise. This group, which I wrote about just a short time ago, already has 60 members and a newsletter that's full of program and book reviews, beginner's series, hardware hints and lots more.

These two groups are examples of what is happening out there. Please don't think I'm bagging the rest – I'm not. However, if your user's group hasn't been listed on these pages before, please write to me (MicroBee Column, *Your Computer*, PO Box 227, Waterloo, 2017), enclosing a copy of your newsletter and all the relevant membership information.

Money For Music

Robotron's Milan Hudacek (he wrote the July article, "Roll Over, Beethoven") has released several new products and a competition to launch them. Here's what Milan has to say:

The 'BeeThoven is a sound generator and joystick interface adaptor for the MicroBee computer. It plugs into the parallel port and converts the 'Bee into a musical instrument, sound-effect generator or an arcade game.

The sound generator comprises three independent voice channels and one noise channel. The generator is fully programmable, including the sound-envelope shape, which enables the 'Bee to be used as a sophisticated professional sound synthesiser.

The joystick interface can be used for the direct connection of two joysticks of the Atari or VIC-20 type. This enables

the user to turn the MicroBee into an arcade game.

Launching this product, Robotron opens a competition, which is going to be held in these two categories:

1: Best Arcade Game Using 'BeeThoven's Joystick Interface And Sound Effects.

2: Best Musical Arrangement Using 'BeeThoven's Music Capabilities.

The competition closes October 1, 1983, and the best three entries of each category will win a full refund of the 'BeeThoven purchase price, plus a 'BeeTalker, which is another of Robotron's products.

The 'BeeThoven, $79 including tax, comes with a comprehensive user's manual and step-by-step tutorial and many program examples. The demo cassette is $5, including tax.

'BeeComposer And 'BeeTalker

BeeComposer is a program which makes full use of the graphics and musical capabilities of the MicroBee. With the 'BeeComposer the user can play Mozart's music as well as the Rolling Stones. The program provides an easy way of composing music, even for the professional. For the beginner, a fun way into musical theory and practice. The program comes with a comprehensive user's manual with many examples, for $29.

'BeeTalker is a speech synthesiser using the phoneme generation principle. It offers an unlimited vocabulary and, with suitable software, higher-than-average speech quality. 'BeeTalker uses the parallel port and comes with a comprehensive user's manual. It costs $99.

Give Milan a call on (03) 720 2173 or write to Robotron at 7 Kiloran Ave, Kilsyth Vic 3137.

Wildcards

A new book called Wildcards is, according to the authors – Messrs Burt, Ford and Nallawalla – a potpourri of application notes and tips for the MicroBee. Quickly summarising its 104 quarto pages of tips and hints, it is not a book that replaces any of Applied Technology's manuals but rather supplements them.

Wildcards Volume 1 (more to come), at $14.95, is good value. Volume 1 contains six chapters and twelve appendices covering topics such as efficient BASIC programming, loading Sorcerer tapes, merging Wordbee files and BASIC listings, music, graphics, games techniques (with examples) and printers and printing. The appendices deal with goodies such as graphics characters and PCG composition aids, screen position conversions and other assorted items. □

By Dom Swinkels

In this month's column, Dom Swinkels looks at a replacement chip for the character ROM, the lightpen and the new micro-floppy disk system.

THE TWO CHARACTER sets of the Hitachi Peach contain two types of Japanese characters that aren't particularly useful to Australians. Available as alternate sets (with ASCII values greater than 128) in the interlace and non-interlace modes, they are stored in a four-kilobyte ROM located underneath the power supply. They can, therefore, be changed by replacing this ROM with another ROM in which the data for a new character set is stored.

Greg Hubbard, of Minerva Microware (9/43 Kensington Road, South Yarra, 3141), has produced such a set, which he calls the Graphics Chip. It provides all the normal characters avaiable on the standard ROM up to CHR$(128) and all the graphics characters in the standard ROM.

The graphics characters are now available in both non-interlace and interlace mode, giving a high-quality appearance to your graphics. In addition, about 70 new characters have been added, including some Greek letters and many more graphics characters, again in both interlace and non-interlace modes. Some of the new graphics characters are designed to allow easy production of large alpha-numeric characters on your screen by printing combinations of these graphic characters.

The chip comes with a demonstration program and instructions for mounting it. Since the character ROM is located underneath the power supply and covered by the RF interference shield, it takes about 15 minutes of careful work to replace the old ROM, but clear and detailed instructions are provided.

However, it does mean that you will only want to change your ROM once or twice, rather than thinking in terms of changing it to suit a particular program. Nevertheless, if you want to do some beautiful graphics, it is worth the $50 price tag.

Incidentally, the replacement chip is actually an EPROM, so virtually any characters can be programmed in. If you need special characters, contact Minerva Microware.

I would like to see a character set with subscript and superscript numbers, such as are used in mathematics and chemical formulas, other mathematical symbols and perhaps a complete set of Greek letters. (The graphics chip contains a partial set.) These would be particularly useful to scientific users of the Peach.

One limitation of these special character sets is that they won't normally print on your printer unless you also have a dot-matrix printer with graphics capability and the software to create the same characters on the printer. Perhaps that's another project for Minerva Microware?

Some Light On The Pen

A good feature of the Peach is the fact that it comes with several interfaces built in, including one for a lightpen. With this in mind, I recently borrowed a pen and had a good look at it.

Since I use a green-screen monitor, I spent several frustrating hours trying to make the pen work on my machine, without success.

I finally found out that it's not supposed to work with a green screen: the light sensor in the pen isn't sensitive to the light from a green screen, and its sensitivity to the various colours on a colour monitor is also variable.

The safest colour to use is white – the pen worked with a black-and-white monitor on my machine and on the colour monitor at my local dealer. On the colour monitor, it seemed least sensitive to red but worked well with all the other colours.

The lightpen comes with some demonstration software on disk (and presumably on cassette if you don't have a disk-drive) and I spent an enjoyable half-hour picking colours and shapes off the bottom of the screen and drawing and painting them on to the top part of the screen on a colour monitor, simply by touching the screen with the pen in various locations. An artist could no doubt create a work of art this way, but for me it was just fun.

Another use of the lightpen is to create programs for easy use without having to have any computer knowledge (for example, completely menu-driven programs), or for a handicapped person who cannot use a keyboard but can control a lightpen. The user can simply pick commands off the screen with the pen, as required.

My summary of the pen is that it works well and is an enjoyable accessory for the Peach. The price is high – $175 plus tax, and the demonstration software is extra – but, then, it is good fun. But, remember, it doesn't work on a green screen.

Small Is Beautiful

The new range of micro-floppies are a real advance in the technology of magnetic storage of data, not because there is anything fundamentally new in them but because they represent a good combination of small advances in the technology.

Announcements of the new micro-floppy drives have been made by various suppliers in the United States, Britain and Japan, covering three or four incompatible formats. However, Matsushita's Panasonic Divison, Hitachi and Maxell do have a single format.

This design is sold in the United States by the Amtek Corporation and advertisements for double-sided, double-density twin drives at US$799 began appearing in the American magazines some months ago. In Australia, Delta was to distribute the equivalent Hitachi product, under the "Pocket Floppy" name, but though some announcements appeared in the computer press, nothing has appeared on the market yet. I'm now assured that units will be available in Australia, through Computer Innovation Group, by the time this issue of *Your Computer* goes on sale.

The drives for the pocket floppies are smaller and lighter than the previous 13 cm drives, and certainly a lot quieter. The only noise is that of a small fan, which runs continuously to cool the system.

The unit I tested contained two single-density drives. The 8 cm floppies come in a cartridge, which measures 8 by 10 cm and is 5 mm thick. The cartridge, therefore, easily slips into your shirt pocket, and can be mailed in a normal-size envelope. (Whether it would survive the postal system is another question.)

The Pocket Floppy requires no separate protective jacket. As it is withdrawn from the drive, two stainless-steel blades close over the reading slots to protect the magnetic surface from dust. It is also easier to store than its larger equivalents and easier to handle. The write-protect tabs are mounted in the case and can be moved in and out, as

required. (The end of peeling off a bit of sticky tape and not knowing where to put it.)

Best of all, the new drives are directly compatible with existing controller cards, and the format and amount of storage are the same as for the 13 cm floppies they replace – that is, nominally 80 kilobytes per side for single-density drives and 160 kilobytes per side if double-sided, giving a total storage of 160 or 320 kilobytes on each Pocket Floppy.

To test the reliability of the drives, I wrote a string consisting of CHR$(1) to CHR$(128) to each of the 560 sectors of one disk. I then copied that from one disk to the other and back overnight, comparing each sector with the original string each time. After 12 hours of this, not a single error had occurred. That was good enough for me.

I then exchanged the floppies in their drives to check drive compatibility and ran it for another 10 cycles of copying back and forth. When no errors appeared, I got bored with this and decided to create some electrical noise to upset my MPI drives. No errors!

The Pocket Floppies should be a real boon for hobbyists and educational and other small users, and for all applications where software or data must be exchanged between different machines. Prices appear to be competitive, compared to the 13 cm drives. I expect a bright future for the Pocket Floppy because of its reliability and convenience, similar to the advantages of cassette-tape recorders over the earlier reel-to-reel recorders. I'll certainly be looking to add some to my system.

Diskette Format

Since I looked at the new Pocket Floppy systems in some detail, it is perhaps worth looking at the format of disk storage. The following comments apply to single-density diskettes, whether they are single- or double-sided and whether they are the new Pocket Floppy or the 13 cm floppy.

Information is stored on disk in 40 tracks, numbered zero to 39. Each track consists of 16 sectors, numbered one to 16, and each sector stores 128 bytes of data. Since track 20 is used by the system as a directory for the diskette, the total useful storage is *39 x 16 x 128 = 79872 bytes* – nominally 80 kilobytes, in reality 78.

Storage of programs and data files is done in units, or "blocks", where one block equals four sectors. Blocks are numbered zero to 155.

The directory track is divided into two parts. Sectors seven to 16 contain the names and other data for each file stored. There are 32 bytes set aside for each file, so there is room for four sets of file-directory information per sector and, hence, for a maximum of 40 files. (The actual number of files may be less if they are large files and, therefore, cause a storage of space on the other 39 tracks.)

Keeping track of storage space on the diskette is the function of the first two sectors of track 20, in what is called the file-allocation table (FAT). Here, details are stored of which sectors are used to store each file. The FAT starts at byte five of sector one and is 156 bytes long, since there are 156 blocks available for storage.

LISTING 1.

```
10 REM..."DISKLOOK"...modified by Dom Swinkels.
20 IF PEEK(29)<>19 THEN NEW ON 15
30 CLS:COLOR7
40 INPUT "ENTER DRIVE,TRACK,SECTOR ";D,T,S
50 A$=DSKI$(D,T,S)
60 PRINT "DRIVE";D;"    TRACK";T;"    SECTOR";S
70 FOR I=1 TO 8:B$=""
80 FOR J=1 TO 16
90 C$=MID$(A$,(I-1)*16+J,1)
100 A=ASC(C$)
110 D$=HEX$(A)+" "
120 IF LEN(D$)<3 THEN D$="0"+D$
130 PRINT D$;
140 IF A<32 OR A>127 THEN C$="."
150 B$=B$+C$:NEXT J
160 PRINT "  ";B$:NEXT I:PRINT
170 INPUT"1=LOOK AGAIN    2=PRINT SCREEN    3=EXIT";S
180 ON S GOTO 40,190,260
190 OPEN"O",1#,"LPTO:"
200 PRINT#1,CHR$(13)
210 FOR X=1024 TO 3023
220 A=PEEK(X):IF A<32 OR A>159 THEN A=32
230 PRINT#1,CHR$(A);
240 NEXT X
250 CLOSE:PRINT:GOTO170
260 END
```

TABLE 1.

```
ENTER DRIVE,TRACK,SECTOR ? 0,20,1
DRIVE 0    TRACK 20    SECTOR 1
00 FF FF FF FF FE FE FE FE FE FE FE FE FE FE FE   ................
FE 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B   ................
C2 1D 1E 1F 20 C2 22 C0 C0 25 C0 FF FF FF FF FF   .... ."..%......
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................

1=LOOK AGAIN    2=PRINT SCREEN    3=EXIT? 1
ENTER DRIVE,TRACK,SECTOR ? 0,20,7
DRIVE 0    TRACK 20    SECTOR 7
44 49 53 4B 4C 4F 4F 4B 00 00 00 00 00 00 21 00   DISKLOOK......!.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4C 49 53 54 49 4E 47 20 00 00 00 00 00 00 23 00   LISTING ......#.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
53 4B 49 50 32 20 20 20 00 00 00 02 00 00 24 00   SKIP2   ......$.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
59 4F 52 43 4F 4D 35 20 00 00 00 01 FF 00 1C FF   YORCOM5 ........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................

1=LOOK AGAIN    2=PRINT SCREEN    3=EXIT? 2
```

your PEACH computer

The code used in the FAT is FF = block available, FE = block used for Disk BASIC. Any other value indicates the block is used for program or data storage. A value C0 to C3 indicates the end of a file and any other value indicates the block number where the next part of the file is stored.

If we have Disk BASIC on our diskette, then it is always stored on tracks zero to two or blocks 00 to 0B (0 to 11 decimal). This is also indicated in the file-allocation table by having FE in the corresponding bytes.

It is, therefore, necessary to always put BASIC on your diskette before you store any other files on it, otherwise some of your files will be written over by Disk BASIC.

To look at the disk format, I have included a version of the DiskLook program in Listing One.

The listing also contains a hardy routine for dumping whatever text is on the screen to your printer. Note that the program requires *NEW ON 15* before you start, to ensure routine work. You may want to experiment with this and determine the change required to print a 40-column page, or any one of a multipage or high-resolution graphics mode.

If you run the DiskLook program and select track 20, sectors one and seven will get something like Table One.

This shows that the disk examined contained BASIC on tracks zero to two and has at least four files stored on it. The name of each file is stored in bytes zero to seven of the 32 bytes set aside for each file. Note that when a file name is less than eight bytes long, the system adds spaces *(CHR$(20))* until the name is eight characters long.

The first two files are BASIC files (00 in byte 11) and the third file is a machine language file (02 in byte 11). The fourth file is a data file (01 in byte 11). Byte 12 indicates whether the file is stored in binary format (00) or in ASCII format (FF). Access to data file may be either random (FF in byte 13) or sequential (FF in byte 15).

Byte 14 gives the number of the block (in hex), at which the file starts on the diskette. In the example in Table One, DiskLook starts at block 21 (hex). If you want to look at the beginning of the file on disk, you must convert this to a track and sector location. Remember that there are four sectors to a block, and four blocks to a track. Block 21 (hex) is, therefore, equal to track eight sector five. This locates the beginning of the file, and the file-allocation table indicates where the next block of the file is located if the file is more than one block long.

To become more familiar with the method of data storage on disk, I suggest you look at some of the other sectors of the directory track and also see what happens when you kill a file. Then look at some program and text files on your disk. See what change occurs to the program as stored on disk when you unlist it or when you save "xxxxx", P to protect it.

For more information on diskette format, with the emphasis on disk-drive systems, as well as other information about the Peach and useful programs, see issue two of *Peach User Notes (PUN)*, available from Micro Innovation Engineers, PO Box 33, Kurrajong Heights, 2758, phone (045) 67-7339. The cost is $2 a copy, or $7.80 for a six-month subscription. □

your APPLE computer

By Peter Sandys

I MUST APOLOGISE for the break in transmission but so many things have been happening that I have not had the time to put finger to keyboard and punch out my column.

It's been a while and many things have happened in the Apple market. Apple has taken over the distribution of Apple products. The Lisa has been released, the Apple IIe, Unofile and Duofile, and hundreds of new software packages, one being the much awaited Ultima II.

I welcome Apple's direct involvement in the market. Times change and only by direct involvement can Apple hope to remain the No 1 personal computer company.

My next welcoming is for the Apple IIe, which is the replacement for the Apple II. With the IIe and Lisa, Apple has once again developed a market leadership. The Lisa is like the second launching of the Apple II – it is a revolutionary machine, with new concepts of computer design.

Finally, Ultima II has arrived and it's been worth waiting for. It is a sped-up version of Ultima I with improved logic and greater adventures. It comes with three disks: Earth, Space and a player disk.

As with Ultima I, the object is to develop your character along the traditional Dungeons and Dragons concept of six attributes. Your aim is to alter the course of history.

The game is played on the surface of the world (yes, this world) in dungeons, towns, castles, villages and space. The extra dimension of different time zones gives the Monty Python effect.

In the year 2111, there was the holocaust in which "ancient civilisations, born of love of beauty and wisdom and thought, turned on one another and, in vicious anger and hate, destroyed almost all of the Earth that had nurtured them." This hatred was caused by the Minax, a dreadful enchantress.

The philosophy of your mission is that by destroying the source of the evil that caused the holocaust, you will alter the course of history.

While different from Wizardry, Ultima II is an exciting and absorbing game – one of the best I have played.

However, there is a bug in the program. It is to do with transportation. The editor has agreed to a free year's subscription to *Your Computer* for the first letter from a reader who correctly identifies the bug. If another bug is found, then the entrant will get some credit in the magazine (but not a credit with the subscription department).

Strategic Simulations

Boy, have they been prolific! I received a copy of Galactic Adventures, a good adventure simulation set in a futuristic time.

It's not up to the standard of Ultima II but entertaining all the same. It is a bit slow and has the drawback of needing either two disk drives or the patience to change disks often.

Computer Ambush

This is an older game which has been compiled so that it runs 40 times faster. It is a straightforward war game in which you control a patrol which is ambushed by the opposing forces.

Set in Europe at the time of the Second World War, it is a classic which has been improved extremely well.

Epidemic

Epidemic is a different type of game in which, as director of the Centre for Disease Control, it is your objective to contain an infection of alien microbes which has struck Earth.

Your job is to oversee the isolation and eventual cure of infected regions. When a region is beyond control, drastic measures are needed. It is within your power to use nuclear weapons to destroy a hopelessly infected area.

Fighter Command

Based on the Battle of Britain, in which the heroic Poms fight off the Colonel Klinks of the Luftwaffe, Fighter Command is a good simulation, but it seems to be written in BASIC. This makes it slow after games like Germany 85.

As the British commander, it is your task to send up patrols, keep bases on alert and keep the morale of your men high.

The graphics displays are good and you can magnify the maps. There is a long and short game mode.

Escape From Rungistan

Leigh Marsland, from South Australia, has written to me, seeking advice on the combination to the lock in the bar in Escape From Rungistan. This was one area that took me some time to work out.

The trap I kept falling into was to put spaces between the numbers of the combination. Enter them as one number/answer.

What's the right combination? I could say that this would destroy the spirit of the game, to avoid admitting that I've forgotten, but I won't. However, I do recall that the most important number was the one in the cash register as it gave the clue to the sequence. Also, I thought there were only three numbers. □

your TRS80 computer

By Rod Stevenson

SINCE MY ADVICE, in an earlier column, that TRS-80 owners should make every possible effort to join a local user's group, I've had a disquieting number of readers complain that their local is "too advanced".

When you think about it, this is exactly what's required: it's no use going to a user group if there is no possibility of learning! Why not raise yourself to your local user group's level, instead of waiting for it to come down to yours?

The Inkey$ Trap

An easily overlooked source of program bugs is the ability of INKEY$ to "remember" a key-press, thus taking something you didn't really want. The advantage is that keys can be pressed in advance; the disadvantage is that an unwanted key-press will register.

There is a location (4099H) which stores the contents of INKEY$, but it's simpler to zero it before input is expected by INKEY$ = "".

Satisfaction

Having seen the new MON-5 monitor advertised by Howe Software in the United States, I couldn't locate an Australian stockist, so I phoned the company direct to find the name.

I was told there is no current Australian stockist. However, I was informed that "we would be most pleased to send you an airmail copy immediately and receive your payment in due course." I accepted, and the package arrived six days later. So I feel it is my duty to pass on to you my satisfaction with this method of doing business. Of course, to expect advance shipment – as I got in this particular case – might be too much, but I've never had a bad deal by going direct to the United States. Of course, it's easier to buy locally, but if you can't find a local agent, don't be afraid to deal direct.

A few short comments on MON-5: it does have extra facilities, but at the cost of now needing two-character commands in place of the single-character ones with which I have become so familiar.

It doesn't show the current memory contents when using the T(ype) command, and I can't find a Symbolic disassembly dump to tape or disk command, as in MON-3 and MON-4. However, the debugging routines make it obviously the one to buy.

Two-Sided Diskettes?

I've not noticed the subject of two-sided diskettes being debated within the pages of *Your Computer*, so perhaps I should start it?

In common with a great many others of my acquaintance, I punch an extra read-protect hole and a pair of extra sector-holes in my diskettes, thus giving two surfaces to use instead of one per diskette. I have a single-head drive, so I still have to turn the diskette over; all it saves is the cost of a new diskette.

The arguments against this practice are that only one side may be good, that the diskette will be rotating in the opposite direction from the other side, and that dirt from the pressure pad may build up on the head.

While I can see the point of all these criticisms, I'll continue until I come to grief. Feedback, anyone?

Personal Help

While I'm pleased to reply to any personal requests for advice, as is John Ross (who I've mentioned in past columns, concerning System-80 problems), we'd both appreciate inquirers enclosing stamped, self-addressed envelopes for our replies.

In John's case, the $5 he asks of non-members of the Adelaide TRS-80 User's Group is well worth it to get, for example, Scripsit to work on your System 80. So please enclose either $10 for membership of our Group, or $5 per problem – I'm sure it's just an unthinking oversight, or perhaps a lack of realisation that the average "fix" takes around two hours.

Also available from the Adelaide User's Group (to members only, though, after John's experience) are customised patches for Scripsit, either tape or disk version, to save to stringy floppy, to drive an Itoh printer, to pause the printer (to allow you to talk on the phone, for instance), to slash zeros, to double-strike for emphasis . . . who knows what can be dreamed up if a need is defined?

Lower-Case Characters

I've mentioned this in an earlier column, but it seems there's a need for a repeat, as I'm getting more than a few inquiries: The Japanese Model 1s had a lower-case character set already installed, requiring only a driver to activate it. Most of the more sophisticated games also have one, as does Scripsit. Or simply *POKE5360,97* and see if you get lower-case. If you do, the solution is obvious.

Even if you don't, it may be that the character generator has lower case, and all you need is the extra video memory. So it's worth trying at least.

If you don't know what I'm talking about, just ask one of the hardware experts at your local user's group to plug in the extra video memory chip before you spend large amounts of money on the whole lower-case package.

System-80 BASIC

Yes, the System-80 BASIC is just like the Model 1 BASIC. It's just that the manuals explaining it don't! But then neither did the early Tandy manuals.

However, there are now a few (newer) volumes available from Tandy, about the learning of Level 2 BASIC. And while you're there, look at the other books Tandy has – some are cheaper than under the original cover.

LDOS Hint

Though the manual says LDOS won't accept such things as DIR and SPOOL into its configuration, if you've got the T(ype ahead) facility operating, you can put these in by typing them in from the keyboard after it says it is configuring and before it has finished.

They're not actually in the configure file, but the effect is the same, and leaves AUTO free for something else.

Just something I discovered by accident. Who knows what may come from fiddling? □

COSMIC SOFTWARE

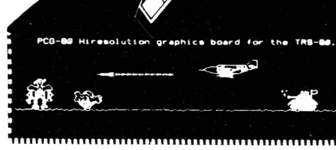
SHARP GRAPHICS SUPER SOUND

Hi-Res Information

THE PCG80

At last, the ultimate hi-resolution modification for the TRS80 Model II & III & SYSTEM 80 has been developed. It provides 256 programmable characters as well as the ability to emulate a 384 x 192 dot addressable screen.

The TRS80 owner can now enter computer graphics at a very sophisticated level. The PCG80 provides hardware capabilities of graphics systems costing hundreds of dollars more. You as an owner can now plot any mathematical function, display arcs, circles, lines, waves. At the same time you can design and display custom characters ranging from 72 pixels to a massive 18432 pixels. You can mix graphics and alpha-numerics anywhere on the screen.



INSTALLATION FEE $30

AVAILABLE FROM

City Personal Computers
75 Castlereagh St.,
Sydney. 2000
Ph: (02) 233-8992

De Forest Software
26 Station St.,
Nunawading. 3131
Ph: (03) 877-6946

Micro Base Computers
422 Newcastle St.,
West Perth. WA. 6005
Ph: (09) 328-9308

Software-80
Shop 8/200 Moggill Rd.,
Taringa, Brisbane
Ph: (07) 371-6996

FEATURES

- The backup is expert. Ask your dealer.
- No conflict with existing hardware. There is no trade off in performance, you do not lose any existing functions. No user memory is affected. The PCG80 is totally user transparent, only active under software.
- As a bonus you get lowercase with full length descenders as part of the mod. That alone must be worth $50!
- It's rapidly becoming the most popular mod to the 80 based machines. hence the software backup will be enormous.
- There is a large library of software available now! It includes both utility software and games. Disk and tape users are supported.
- It's easy to program, no sophisticated programming skills are needed to get it to work. It comes fully documented with both the beginner and expert programmer catered for. You can program in hi-res within 20 minutes of installation. Convert existing software to hi-res in 60 minutes.

***TRS-80 MODEL I = $199 SYSTEM 80 MK I AND BLUE LABEL = $219**
ON SITE INSTALLATION $30 + SERVICE CALL SYDNEY METROPOLITAN AREA

* Please specify TRS80, US keyboard or Japanese keyboard.

HRBASIC

A complex extended Basic, HRBASIC triples the power of Basic and your PCG. Far superior to other high resolution languages currently available. HRBASIC offers maximum speed and pixel coverage on a 384 x 192 dot addressable screen. Commands include HPLOT which allows you to set and reset or draw lines instantly. Other commands include: HCOPY DMODE, PROGRAM, HELP, HNXT, DMODE, HINIT, HBLOCK, HRES, HSAVE and more!

16K Tape — $29.95 48K Disk $49.95

SUPAEDIT

The ultimate character editor. Has all the features of HREDIT and more. SUPAEDIT writes portions of BASIC code (data statements) and assembler source that can be incorporated directly into your own program. A fast and fantastic way to design complex high resolution images instantly and easily.

48K Disk only $49.95

COSMIC SOFTWARE'S PCG80 GAMES

Starbird	16k tape $19.95	Killer Beetles	16k tape $19.95	PCG Trek	16K tape $19.95
Star Cresta	16K tape $19.95	Road Patrol	16K tape $ 9.95	Jungle Raider	16K tape $19.95
Ghost Hinto	16K tape $19.95	Centipede Attack	16K tape $19.95	Alien Cresta	16K tape $19.95

HOW TO ORDER:
Order the products you want on our COSMIC HOTLINE! Ph: (02) 661 4075. Use your Bankcard. Or order direct through our superfast Mail Order Department! See order form. If you do not want to cut it out then just copy out the main details! All orders are sent out to you within 24 hours of receiving your order by CERTIFIED PRIORITY PAID MAIL.

DEALER ENQUIRIES WELCOME.

•Post to: COSMIC SOFTWARE
G.P.O. Box 3494, SYDNEY N.S.W. 2001
Phone: (02) 661 4075

NAME . . . .

ADDRESS . . . .

. . . . P/CODE . . . .

Enclosed$. . . . . Bankcard/Cheque/ Money Order

Expiry Date:. . . . .

Bankcard Number . . . .

Signature . . . .

ITEM	QTY	$ PRICE EACH c	$ AMOUNT c

ALL PRICES INCLUDE POSTAGE! TOTAL________

By Bill Bolton

MICROPRO'S WordStar word-processor is now a widely used package. I use WordStar extensively for a wide range of text processing, most of which is intended for screen display rather than hard-copy printout.

While WordStar can give nicely formatted files for screen display when used in the D(ocument) mode, it can generate some strange effects on many VDUs, if a WordStar-formatted file is displayed with the CP/M TYPE command.

This is because WordStar sets the Most Significant Bit (MSB) on many characters in the text to mark them as having a special meaning to WordStar. The last character in every word has the MSB set to show the end of a word for line-break purposes, and so on.

Another example is the "soft" carriage return at the end of a line that WordStar may need to re-format at some time. This is a normal ASCII carriage return (0DH) with the MSB set (making it 8DH). There are numerous other characters which WordStar may set the MSB on.

Many WordStar users have discovered that running a WordStar file that has the MSB set on some characters through PIP with the "Z" option will set the MSB to zero on all characters and create a file which has only valid ASCII 7 bit text in it. For example:

PIP MYFILE.TXT=MYFILE.WST[Z]

This is a simple approach using a standard CP/M utility available to all CP/M users. It still doesn't solve all the problems, however – there may still be a few "time bombs" left in the resulting text file. Some of the characters on which WordStar may set the MSB turn into ASCII control characters when the MSB is set to zero.

Additionally, when the "soft" hyphen is in the middle of a word and not at the end of a line, WordStar gives it a value of 1EH in the file. On many terminals, this is the "home cursor" command and, if encountered in a text stream, will send the cursor to the top of the screen and start writing over the text already on the screen.

The printer-control commands ^B, S and so on are also still in the document after this PIP operation. While these do nothing unusual to my terminal, they can create havoc on others when the file is TYPEd.

To overcome this problem, I took a simple utility from the BDS C users group and did further work on it to create a WordStar file scrubber that will "scrub" all the strange things out of a WordStar text file with embedded format information to make it into a safe ASCII text file. The program changes soft hyphens at the end of lines into hard hyphens and removes all control characters from the file, except for carriage return, linefeed and tabs.

Most C programmers would class this program as a "filter", in that it takes the output from one program (WordStar) and does something to it (filters it) to make it suitable for input to another program (TYPE).

The version shown here is for the Computer Innovations C 86 compiler. A version for the BDS C compiler is available in the SIG/M users group collection (you can request it for downloading on the ST-RCPM or SPA-RCPM systems).

I have also compiled this code under the Lattice C compiler for MSDOS and it required only a couple of trivial changes to compile correctly.

Australian CP/M User Group

Peter Harris, of Canberra, is keen on starting the Australian CP/M User Group, and has already registered the name. His idea is to set up a formal organisation, with a constitution, office bearers and so on.

Now, I can't really get turned on by the thought of a "formal" organisation, especially as the whole concept of the United States user groups is to avoid formal organisation as far as possible (which, on the whole, seems to have worked okay so far). However, I do get many queries from CP/M users around Australia who feel they would like to "join" a CP/M user group of some sort.

If you're interested in Peter's proposal, you can write to him care of the magazine – we'll pass it on.

If you expect a reply, you should include a self-addressed, stamped envelope as a matter of courtesy.

Personally, I see RCPM systems as being a more effective way of getting CP/M users together and distributing public domain software, as they neatly bypass the problems of disk formats, newsletters, meetings and "formalities", and so on.

By Way Of Explanation . . .

A couple of things. First, in the Christensen Protocol article in the June issue, an ASCII "ACK" was defined as 05H, but – as any good hacker knows – it is really 06H (so how come only one reader spotted the error?).

Second, the local agent for Holt, Reinhart and Winston, publisher of the book *Inside CP/M*, reviewed last month, is Holt Saunders Pty Limited, 9 Waltham Street, Artarmon, 2064, phone (02) 439-3633 Sydney.

Digital Packet-Radio

There's an increasing amount of interest – in the Sydney area, at least – in digital packet-radio. A Sydney amateur-radio group is now publishing a quarterly newsletter, called *Australian Packeteer* which covers developments in the field.

A special "packet" processor board is available at reasonable cost to implement the "digital packet" part. As good commercial two-metre radio equipment is readily available, it is certainly possible to get a digital packet-radio interface on to a computer system without having to know a lot about the radio side.

The digital packet-radio scheme was developed in Canada and uses HDLC protocol. An RCPM system with packet-radio access should soon be available on 2 metres (147.6 Mhz) for those in the Sydney/Newcastle/Wollongong area, a repeater is planned for Berowra which should give good coverage.

For more information on digital packet-radio, contact Jim Swetlikoe on (02) 452-2668 Sydney (reasonable "out-of-business" hours only, please) or on call sign VK2BVD. Jim is editor of *Australian Packeteer.*

Masses Of Volumes

Software Tools' RCPM system has recently added a further two volumes to the MISC software collection, bringing it up to 25 volumes. Volume 24 is locally contributed software while volume 25 is CP/M-86 utilities.

Also, by the time you read this, ZCPR version 2 should be available from at least some of – if not all – the RCPM systems.

The SIG/M user group collection is now up to volume 117. I have not had any information of new volumes (past volume 90) from the CP/M user group for some time.

PAMS Numbers

The number of public-access systems has grown now to a point where it's worthwhile listing the numbers each month. So, for the time being at least, look at the end of this column each month for a list of the systems which allow full or partial free access. This month's list:

Mi-Computer Club BBS (MiCC-BBS): (02) 662-1686, 24 hours.

Software Tools RCPM (ST-RCPM): (02) 997-1836, 24 hours.

Micro Design Lab RCPM (MDL-RCPM): (02) 663-0151, 5pm to 7am.

Sydney Public-Access RCPM (SPA-RCPM): (02) 808-3536, 24 hours.

Melbourne CBBS (MICOM-CBBS): (03) 762-5022, 24 hours.

Gippsland RCPM (GL-RCPM): (051) 34-1563, 24 hours. □

```
/*
        Program to "scrub" a WordStar text file back to a
        standard ASCII file.

VERSION LIST, most recent version first

15/Mar/83
        Massaged source to suit Computer Innovations C 86 compiler
        under CP/M-86 or MSDOS (should also work for other compilers
        using the K&R syntax such as Lattice C under MSDOS etc.).
        This mostly consisted of changing file I/O statements to
        agree with Kernighan and Ritchie syntax. Changed counter
        variables to "long" so that large file sizes are reported
        correctly. Removed output file name check until I have time to
        do an extensive recoding. Bill Bolton

26/Sep/82
        Forces MSB of all characters to 0, then scans for control
        codes. TAB, CR and LF are passed unchanged to the output
        file. US (soft hyphen) is replaced by a hard hyphen.
        Checking for legal CP/M filename on destination file
        added. Expanded "usage" message. Added "working" messages.
        BDS C version for CP/M-80.
        Bill Bolton.

        This program was developed from a program called SCRUB
        on BDS "C" User Group disk "Utilities 2" (Volume 2 in
        the Software Tools RCPM BDSCAT.ALL).
*/

/*
        Macros for constant definitions
*/

#include <STDIO.H>

#define VERSION 1       /* main version number */
#define REVISION 1      /* sub version number */
#define DEL 0x7F        /* ASCII delete character */
#define WORKING 1024    /* number of chars between progress markers */
#define NEXTLINE (WORKING * 32) /* number of progess chars on a screen line */
#define CPMEOF 0x1A     /* CP/M-86 end of file marker */
#define ERROR 0         /* Normal file error condition */
#define FERROR -1       /* Flush file error */

/*
        Argument vector indices
*/

#define FROM_FILE 1
#define TO_FILE   2

/*
        main to open the files for scrub()
        and handle invocation errors.
*/

main(argc,argv)
int argc;
char *argv[];

{
        char *fdin,*fdout;
        char buf[12];

        printf("\nWordStar file Scrubber Version %d.%d\n",VERSION,REVISION);
        printf("Bill Bolton, Software Tools\n");
        if( argc != 3 )
                usage();
        else {
                if( (fdin = fopen(argv[FROM_FILE],"r")) == ERROR){
                        printf("\nCannot find file %s\n",argv[FROM_FILE]);
                        usage();
                }
                else {
                        if( (fdout  = fopen(argv[TO_FILE],"w")) == ERROR )
                                printf("\nCan't open %s\n",argv[TO_FILE]);
                        else {
                                printf("\nWorking ");
                                scrub(fdin,fdout);
                        }
                }
        }
        exit();
}

/*
        procedure scrub -- copy file to file deleting unwanted control chars
*/

scrub(fdin,fdout)
FILE *fdin;     /* the input file buffer */
FILE *fdout;    /* the output file buffer */

{
        int c;                          /* 1 char buffer */
        long count;                     /* count of characters processed */
        long killed;                    /* numbers of bytes deleted */
        long hyphen;                    /* number of soft hyphens replaced */

        count  = 0;
        killed = 0;
        hyphen = 0;

        while( (c = getc(fdin)) != EOF  && c != CPMEOF ){
                c &= 0x7F;
                count++;
                if (count % WORKING == 0)
                        printf("*");            /* still alive */
                if (count % NEXTLINE == 0)
                        printf("\n\t");         /* new line every so often */
                if( c >= ' ' && c < '\177' )    /* visable character ? */
                        putc(c,fdout);

                else
                        switch(c) {
                                case '\r':
                                case '\n':
                                case '\t':
                                        putc(c,fdout); /* ok control chars */
                                        break;

                                case '\037':    /* replace WS soft hyphen */
                                        putc('-',fdout);
                                        hyphen++;
                                        break;

                                default:
                                        killed++;
                                        break;          /* ignore it */
                        }
        }

        putc(CPMEOF,fdout);                     /* sent textual end of file */
        printf("\n");

        if( fflush(fdout) == FERROR)
                exit(puts("\nOutput file flush error\n"));

        printf("\n%ld characters processed\n",count);
        printf("%ld characters were deleted\n",killed);
        printf("%ld soft hyphens replaced\n",hyphen);
}

usage()

{
                printf("\nUsage:\n\n");
                printf("\tSCRUB d:file1 d:file2\n\n");
                printf("Where:\n");
                printf("\tfile1 = source file, (* and ? not allowed)\n");
                printf("\tfile2 = destination file, (* and ? not allowed)\n");
                printf("\td:    = optional drive identifier\n\n");
                printf("i.e.\tSCRUB A:FOOBAR.WST B:FUBAR.DOC\n");
}

/* end of scrub */
```

Free readers' classifieds*

FREE CLASSIFIEDS: are for readers only, not commercial organisations. Classifieds of 20 words or less accepted free of charge; for classifieds of more than 20 words the first 20 words are free, then it's 20 cents a word.

Black and white prints may be included if suitable for publication, and will be charged for at $10 each photograph.

More than one ad in one issue will be treated as a single ad and charged accordingly.

COMMERCIAL RATE: Minimum 20 words, at 35 cents a word. Photographs $20 each.

PRINT or type your ad clearly and legibly, double-spaced, and include separately your name, address and phone number for our records or checking purposes even if these details aren't all to be included in your ad.

Send it to *Your Computer* Readers' Classifieds, PO Box 21, Waterloo, NSW 2017.

For Sale: Centronics 101A printer, up to 132 columns, tractor feed, fast, parallel connections, presently used with TRS-80, phone Alan, after 7pm, (02) 623 6403.

Microsoft Fortran 80: To suit model 1 TRS-80 or System 80 single or multiple disk. Two diskettes and manual $150. Ring (075) 39 4895 (John) nights.

CBM-64: 2 disk drives, printer, 2 months old accept $2300, will separate. Phone (02) 349 8521.

Sell: Apple II, CP/M, Green screen, extra software $3500 ono. Osborne I, single density, extra software $1800 ono. (02) 818 1062 ah.

Microbee: Owners need printer's mate to make full use of a dot matrix printer. Machine language utility provides page formatting, graphics screen dump, text screen dump, memory dump. Available for Itoh 8510 and Epson FX80 printers. Enquiries invited for other printers. $16.50 includes programs, instructions, examples, packing and postage. Icarus Software, 8 Maplin Place, Rossmoyne, W.A., 6155.

For Sale: VIC-20 computer, datasette, joystick, $300 worth of games, software, books, magazines, etc. Sell $650. Phone (03) 578 6942.

Manual: 25 ways to make your computer pay for itself. Send $4 to J S Foster, Linthorpe, Ongerup 6336.

For Sale: One 12 inch diagonal Sanyo green screen monitor, new and unused in original packing. Separate brightness and contrast controls, 18 MHz bandwidth. Phone (02) 660 9316 or 519 2660. $200.

User Group News: Chip-8 users are still catered for and all are welcome to join. Whether your computer is an RCA VIP, an ETI 660, or the Dream 6800 you are welcome to join us. If you have the new COMX 35 and want to find somebody who speaks 1802 please write. Our languages are Chip-8, 6800, 1802 – from September issue of our newsletter 'Dreamer' many new items of special interest shall be printed. For details write to Frank Rees, 27 King St, Boort, 3537.

Must Sell: ZX81 16K – adaptor, leads, manual, books, many programs, games and commercial software. Value $400, sell $330 ono. 56 Ann St, West Geelong, 3218.

ANSUA: Australian North Star Users Association are pleased to announce the formation of their club. Further details are available from ANSUA, PO Box 194, Wangaratta, 3677.

HP 75-C: $1200. Cassette drive, suit 75-C and 41CV $450. HP 9" monitor & interface also suit 75C & 41CV $500. All with manuals and as new. (03) 870 2161 or write to J Dawson, 37 Jarma Rd, Heathmont 3135.

Super 80 Users: Large readable circuit diagrams. 1000 mm x 707 mm accurately redrawn to board. Ideal for servicing. $13. Brian Morris, 57 Kangaloon St. Jindalee, 4074.

Apple Software: On DOS 3.3, James McRae (02) 872 3086 ah.

Kaypro II: As new, software and manuals still sealed. $2800. GBS Data Manager Package for Kaypro $250. Phone (03) 579 3430.

How To Buy a Personal Computer: This book explains what you need to know and gives comparisons of 24 popular PCs. Send $6.95 Chq/MO to Ozway Consultants P.O. Box 1639, Nth Sydney, 2060.

Votrax Type-N-Talk: Text-to-speech synthesizer, unlimited vocabulary, built in amplifier. Complete with manual, power supply, cable and driver program. Cost $500. Sell $300. Phone (02) 44 1126.

MicroBee Software: Educational software suitable for use in secondary school classrooms. For available programs contact G. Clark, 147 Gibbons St, Narrabri, 2390. Ph (067) 92 1716.

For Sale: Atari TV game with 14 game cartridges, (Defender, Starmaster etc.) includes every type of controller. Sell $495. Phone (077) 74 0260.

A.P.F. Users Group: Now formed and looking for additional members for exchange of news, software, technical information, etc. Contact Norm McMahon (02) 44 2645 or write 288 Kissing Point Rd, Turramurra, 2074.

TRS-80: 16K Extended BASIC colour computer. Includes joysticks, $110 worth of software, manuals and cassette cable. Worth $850 for $699. Phone (059) 56 7233 ah.

MicroBee Software: 'Stock Supervisor' provides a method of stocktaking which can be updated periodically, or as stock is received and issued. Each product has two price fields (e.g. wholesale and retail). Percentage margins and money totals on each item are calculated, together with grand totals for both price entries and average product margin. The optional printer routine allows for the printing of separate stocktaking reports, price lists and inventory schedules. Tape with full instructions $15. Mail order to B Proctor, 27 Roberts Avenue, Mortdale, 2223. Prompt return guaranteed.

Beefile: A file program for the MicroBee suitable for 16 and 32K machines with tape recorders. Cost only $9.50. Tel (02) 628 0324 for further information, after 5pm.

BBC Model B 32K: Sacrifice at $1200 only. Unwanted prize. Phone (02) 798 3379 ah.

MicroBee: 16K adventure 'Lunar 7' send $10.50 to J Baker and D Jacobs, 8 Gould St, Frankston, 3199. Price includes postage.

MicroBee Froggerplus: A BASIC game for your MicroBee! Get past the cars and trucks then get across the turtles and logs – you're home! On the same tape is Cosmic Pilot, destroy the aliens before your fuel runs dry and they invade Earth. Send only $6 to Tom Caton, 3 Baust Cres, Coromandel Valley, 5051.

Atari 800: 48K, 410 recorder, 2 joysticks – $1000. Atari 810 disk drive, $700. Over $1000 worth of software and books available. All hardware and software for $2100. Ph (02) 232 6366 bh or 449 6286 ah.

Hawkesbury MicroBee Computer Club: meets at 7.30 pm on the first Friday of each month, at Richmond Primary School (NSW). With an average attendance of 30 people the club looks to support local education and business on a non-profit basis. For further information contact the President, Mr Bruce Rennie, on (045)67-7329.

MicroBee: Forth for the MicroBee, AFIG Forth ready to run on your 32K 'Bee. With cassette routines and a screen editor, this language is truly remarkable and fast. Includes one year membership of the Australian Forth Interest Group, cassette and comprehensive manual. Cost $45.00 + $1.00 p&p, send cheque to Tony Latemore, PO Box 704, Sale. Ph. (051)44-2011.

ZX81: 1K, leads, manual, 1.2 A power supply. Numerous programs, construction articles (add-on memory etc). Just $170 ono. F. Los, 13 Kroombit St, Dulwich Hill 2203.

Terminet 1200: Printing terminal, RS232-C, 1200 baud, U/L case, spares, VGC, $350. Also teletypes ASR33, $150, and teletype KSR35, $200 – all come with manuals. Ph. (03)553-3428 ah.

ROM Extension Card: Plugs into coreboard bus socket, allows four set of EPROM in EDASM and three set in net location. All EPROM software selectable. Includes 8 I/O inputs and 11 outputs (open collector), comes complete with range of software and comprehensive manual. Kit $120.00 assembled $180.00. F M Capmeil, 123 Hargrave St, Paddington, 2021, (02) 328 1246.

market directory

DEALERS, distributors, mail-order houses and other commercial organisations or individuals who have a message to spread but don't want to outlay the sums required for full-scale advertising can take advantage of our Market Directory — small ads to help those searching for outlets, services and equipment. For details contact *Your Computer* on (02) 662-8888 or write to PO Box 21, Waterloo 2017.

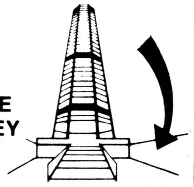
---

the **Computer Spot**

TRADE IN YOUR VIC-20 FOR COMMODORE 64 NOW!!

Call in or phone for a special deal

SHOP C4, MLC CENTRE
MARTIN PLACE, SYDNEY



PHONE: (02) 235 2971

JANUS is a subset of the famous **ADA** language.

JANUS contains everything you need to do fast structured program development in a micro environment.

CPM-80 and CPM-86 versions and MSDOS versions available.
Prices from $400.

MICRONIX Phone (07) 57-6999
11 Blackmore St., Windsor, Qld. 4030
Dealer enquiries also welcome

ROCKSOFT SOFTWARE **ZX81**

BLANK TAPES – C15
8 FOR $9.99
ZX BOOKS ALSO AVAILABLE
ALSO
CATACOMB
3D DEFENDER
3D MONSTER MAZE
AND MORE $14.99
NEW-ALIBI
A super game of logic and deduction for your ZX81/SPECTRUM.
SPECTRUM SOFTWARE NOW AVAILABLE
PRICES INCLUDE POST & PACKAGE
BANKCARD WELCOME
SAE FOR CATALOGUE

ROCKSOFT
G.P.O. BOX 5194 AA, MELBOURNE 3001
ORDER HOTLINE (03) 63 1886

HITACHI PEACH

FORTH

$50 buys qBITS FORTH (interpreter/compiler, plus screen editor). A structured, modular language 20 times faster than BASIC.
Documentation includes comprehensive FORTH glossary. Single or double density disk — please specify which one you require.

qBITS
P.O. BOX 557,
BELCONNEN,
A.C.T. 2617

ATTENTION PROGRAMMERS

Market Your Software Internationally

COMMODORE 64 — VIC 20
ATARI 400/800 — I.B.M.

If you have written, or intend to write saleable software then let us market it for you both here in Australia and overseas. Our royalty rates are extremely generous. If you are interested then submit a sample for evaluation, or phone us during business hours.

MICRO INTERNATIONAL
G.P.O. BOX 2427,
ADELAIDE, S.A. 5001
AUSTRALIA
TELEPHONE: (08) 79 9951 TELEX: 87448

HUGHES Phototype

TYPESETTING BY TELEPHONE

FROM COMPUTERS OR WORD PROCESSORS

CHEAPER • FASTER • ERROR FREE
2 MACPHERSON ST, CREMORNE, NSW 2090
90 4034

Molymerx is the largest source of TRS-80 software in Australasia

Send **$3.00** for the **Molymerx Software Catalogue**

- 250 programmes • 80 pages
- quarterly updated information on our new releases, specials & hints

Post orders to:
Molymerx Pty Ltd,
P.O. Box 900, GOSFORD, 2250.
OR Phone us on:
(043) 694 888

ENTERPRISE Computing
Software for Microbee.

Can you resist the challenge of these superb adventures?
LOST ISLAND ADVENTURE—$14-45
THE MARIANIS TRENCH—$14-45
Both programs utilise superb graphics and sound effects.
This is top quality software from an Australian company. Write or phone for catalogue *or* order now for prompt delivery.

Bankcard Welcome

Enterprise Computing
4 Fintonia Street
Hughesdale, Vic., 3166
Ph: 579-4104 A/H 380-5634

COMPUTER DYNASTY

A WIDE RANGE OF COMPUTER PERIPHERALS AND SOFTWARE FOR BUSINESS, HOBBY, EDUCATION AND ENTERTAINMENT.

15 FLORENCE STREET, HORNSBY, 2077
(At foot of Rail overpass)
PH (02) 477 6886

LOW-COST **ZX81** PROFESSIONAL
SOFTWARE

Huge range of fully guaranteed:

- **GAMES** • **UTILITIES**
- **BUSINESS** • **APPLICATIONS**

To immediately receive your FREE catalogue, send a stamped, self-addressed envelope to:

omnisoft
428 NELSON ROAD, HOBART, 7007

---

16 BIT/16K RAM/16 COLOUR & 5 OCTAVES SOUND
ALL FOR $499

TEXAS INSTRUMENTS TI 99/4A

***LANGUAGES:** TI-BASIC (STANDARD), EXTENDED BASIC, EDITOR /ASSEMBLER, USCD-PASCAL, TI-PILOT, TI-LOGO II

***APPLICATIONS SOFTWARE:** FULL RANGE OF
- BUSINESS/HOME/PERSONAL PACKAGES
- EDUCATIONAL (STUDENTS & TEACHERS)
- OF COURSE, COLOURFUL GAMES
- ALSO MICROSOFT MULTIPLAN — SPREADSHEET
  TI-WRITER — WORD PROCESSOR

***OTHER: SPEECH SYNTHESIZER, PHONE-MODEM COMMUNICATIONS, PARALLEL CENTRONICS PORT, RS232 SERIAL PORTS, TAPE DRIVES, DISK DRIVES, ETC.**

MICROKEN
833 VICTORIA ROAD, RYDE 2112 (02) 808-3259

your computer services

LETTERS TO THE EDITOR: We are happy to receive your comments and, if they are of interest to other readers, publish them. Letters will only be considered for publication if they include your name and address, although we can withhold such details from publishing on request. Note that we reserve the right to (and probably will) edit all letters for the sake of brevity, clarity or accuracy.

SUBSCRIPTIONS: Standard rate within Australia, $24; airmail $35. Available free with membership to the Mi-Computer Club (club membership fee of $30 includes free magazine binder and sticker). All overseas rates are a minimum $10 (surface) or $15 (air) higher, are quoted in (and must be paid in) Australian dollars; they are available on application. Allow up to eight weeks for subscription processing.

BACK COPIES: and single issues are available from the publisher's office ($2) or by mail ($2.50). We will supply photostat copies of articles where a back issue is not available, at the single-magazine price for each feature copied.

READERS' ENQUIRIES: We will make every effort to answer readers' written enquiries, if accompanied by a stamped, self-addressed envelope, although staff shortages and deadline pressures may cause delays. **Please include your telephone number(s) with any enquiry**. Phone enquiries not related to subscriptions, readers' advertisements, or other 'service information' cannot be accepted.

LIABILITY: Although it is policy to check all material used in *Your Computer* for accuracy, usefulness and suitability, no warranty, either expressed or implied, is offered for any losses due to the use of any material in this magazine.

EDITORIAL CONTRIBUTIONS: Contributions to Your Computer are welcomed and will be given every consideration*. Please read these notes carefully to get an idea of the style and format we prefer.

All Contributions: should include your name, address, and home and office phone numbers (in case we need to check details). Each page of your submission, and any material sent with it, should also carry your name.

Contributions on disk: Contributions can be accepted in a wide variety of disk formats, although some we have converted outside our offices, which will add to the (often lengthy) delay between receipt and acknowledgement. The preferred medium is IBM standard format single-sided, single-density 20cm CP/M disks. We can also handle, in-office, Kaypro II and Osborne 13cm disks, MicroBee tapes and 13cm Apple DOS or Apple CP/M disks. If you have one of the formats listed by Archive Computer Services in Brisbane (usually advertised in this magazine), we can accept that too. Please pack them extremely carefully if posting and label all disks with your name, address and phone number.

Contributions By Telephone: Contributors who have modems and suitable software (in the MODEM7/YAM mould – see our stories on Christensen Protocols in the May and June 1983 issues) can arrange direct transfer to our computers, or obtain special contributor access to the Mi-Computer Club Bulletin Board system, which is on-line 24 hours a day, seven days a week.

Style: All items should be typed (or printed) and double-spaced on plain white paper. We will only accept original copies – no photostats. Include your name, address, telephone number and the date on the first page of your manuscript (all manuscript pages should have your surname and page number in the top right-hand corner). Be clear and concise, and keep jargon and adjectives to a minimum.

Listings: Unless it is absolutely impossible, we want listings produced on the computer. This reduces the risk of error – if the computer typed it, the computer probably accepted it. Print listings with a dark ribbon on white paper, and try to format the output to a narrow (40-characters) width. If they can't be produced on a printer, borrow a good typewriter – hand-written material is likely to sit around the office for a year before someone can find time to type it all out for you! Please provide an account of what the program does, how it works and so on. Any comments on the program should refer to the address, line number or label rather than to a page number. Any comments on modifying the program to work on other machines will be appreciated. Try to include a printout of at least part of a sample run if possible.

*Although the greatest care will be exercised with contributions, no responsibility can be accepted for the safety or return of any letters, manuscripts, photographs or other materials supplied to *Your Computer* magazine. If return is desired, you should include a stamped, self-addressed envelope. If return is critical – say it's something you can't afford to lose – then don't send it; we are careful, but we're not perfect, and Murphy's Law guarantees yours is the one item we'll lose this year.

advertisers index

1984
THE POSTER—THE CALENDAR

BY PUBLIC DEMAND:

Ideal for the office or the study, **Your Computer's** 1984 wall-poster calendar measures 43 x 56.5 cm, and is printed on quality gloss art paper. Featuring the artwork used for our June issue's cover story, "The Next Generation", the calendar costs just $3.00, including postage and packaging.

Fill in the coupon today to avoid disappointment.

Calendar Offer,
Your Computer,
PO Box 227,
Waterloo, NSW 2017

Please rush me .... copy/copies of your 1984 wall-poster calendar, at $3.00 each (including postage and packaging). Enclosed is my cheque/money order for $..............

Name ....

Address..

.... Postcode..................

INTRODUCING MOSQUITOES.

Write for a free catalogue

A sparkly new dimension in computer games from an all-Australian programme.

If Harrison Ford had you at the edge of your seat, when he was chased by boulders and poison darts; if the tense excitement of seeing him trapped in the ancient Egyptian ruins, in search of the lost ark, with all those serpents left you breathless; then this is going to be a real treat for you.

You're entombed beneath massive stone walls. Your only escape is through the locked door. You see the key, but there are swarms of deadly mosquitoes. You do have some insecticide so you may get courageous, but if they strike from above or below you are defenceless. Is there no escape? Yes! The ventilation tunnels. But where do they go? You won't find that out until you come out the other side.

So the heart-pounding race is on and if you have the nerve and skills of a true adventurer you may make it through the first door, only to find you're now in the dark. The key. You see the key, but finding your way through these secret chambers, evading the mozzies, and making it through the next door will leave you faced with the ultimate challenge.

This very original experience of electrifying graphics and startling sound will blow your mind, now, that the danger of the mosquitoes is compounded with the mystery of locating the key ... and we can't tell you where to find it because it doesn't play the same way twice, and you've never had as much fun with a VIC 20 until you take up the challenge of Mosquitoes!

AVAILABLE THROUGH YOUR COMPUTER DEALER

PARATROOPER

Realistic smooth action and true hi-res graphics. You are the only one left to stop them. The sky is full of enemy choppers. Paratroopers keep dropping into your area with a non-stop barrage of enemy troops. They are out to destroy you. This new game is an unbeatable blend of arcade action and quick thinking strategy. You must make every shot count — don't be too fast on the trigger. Every time you hit a chopper or paratrooper you get extra points. Wait until you see the climax of this game — you won't believe it! This is a multiple skill level game with razor-sharp graphics and sound. Joystick or keyboard runs in standard VIC 20.

ROAD TOAD C64/VIC 20
A new and unique version of Frogger, now a multi-action game for both computers. $19.95 VIC 20; $24.95 C64.

CHICKEN CHASE C64/VIC 20
Collect all the eggs from the chicken coop without being killed by the fox. $19.95.

UNDERWORLD OF KYN C64
Destroy the powers of the evil wizard of Kyn and escape from the twisted underworld. A challenging real-time graphics adventure for the C64. $24.95.

ANNIHILATOR

Defend your moon base in this fast-pace, high-action game against attacking aliens who launch sophisticated heat-seeking missiles at you. Your Commodore 64 will amaze you with the complexity of sound and graphics in this programme. A similar programme is also available for VIC 20 and they'll provide hours of entertainment for everyone, or just about everyone because we all can't have a keen eye and a fast wit!

BUG-BLAST

The 'get-them' before they 'get-you' fun of Bug Blast is superb entertainment for the young and the computer novice. If you thought Centipede was fun, then shooting through the cactus with the hit-and-miss action of Bug Blast is going to inspire your imagination. Available for VIC 20, and Commodore 64.

DEALER INQUIRIES INVITED. TELEPHONE

(02) 29 6330

IMAGOMAT 1016